National Defence / Défense nationale

UNCLASSIFIED

UNLIMITED DISTRIBUTION

5

ADA133006

DRES

SUFFIELD MEMORANDUM

NO. 1081

A NONLINEAR SIX DEGREE-OF-FREEDOM BALLISTIC AERIAL TARGET SIMULATION MODEL (U)

by

A.B. Markov

PCN No. 21V10

August 1983

DEFENCE RESEARCH ESTABLISHMENT SUFFIELD : RALSTON : ALBERTA

Canada

WARNING
The use of this information is permitted subject to recognition of proprietary and patent rights.

UNCLASSIFIED

UNLIMITED
DISTRIBUTION

DEFENCE RESEARCH ESTABLISHMENT SUFFIELD
RALSTON ALBERTA

SUFFIELD MEMORANDUM NO. 1081

A NONLINEAR SIX DEGREE-OF-FREEDOM BALLISTIC AERIAL TARGET SIMULATION MODEL (U)

by

A.B. Markov

PCN No. 21V10

~~**WARNING**~~
~~"The use of this information is permitted subject to recognition of proprietary and patent rights".~~

UNCLASSIFIED

ACKNOWLEDGEMENT

The software documented in this report was coded, debugged and tested by Mr. K. Schilling of West Germany. Mr. Schilling was a visiting NATO fellow to DRES in the period September, 1980 to December, 1981.

DEFENCE RESEARCH ESTABLISHMENT SUFFIELD
RALSTON ALBERTA

SUFFIELD MEMORANDUM NO. 1081

A NONLINEAR SIX DEGREE-OF-FREEDOM BALLISTIC AERIAL TARGET SIMULATION MODEL (U)

by

A.B. Markov

ABSTRACT

Six degree-of-freedom, rigid body equations of motion are described suitable for modeling the dynamic characteristics of multistaged, free-flight, ballistic rockets such as the DRES developed aerial targets CRV7/BATS and ROBOT-9. These equations of motion form the core of a FORTRAN simulation software package called BALSIM. This package allows for modeling of vehicle thrust and structural asymmetries, time-varying mass and inertia characteristics, variable wind conditions, nonstandard atmospheric conditions, stage failures, and different rocket motor types. The BALSIM package has been written in IBM FORTRAN IV and has been tested on the IBM 3033 computer with the H-extended compiler. It is currently being adapted for use with the VAX11/780 and Honeywell DPS-8/70C computers.

(U)

TABLE OF CONTENTS

Page No.

LIST OF FIGURES

LIST OF SYMBOLS

The following is a list of important symbols. Some symbols, which are defined in the text and are used only once, or are secondary quantities related to a primary quantity that is apparent from the text or from the notation conventions to follow are not included. Numbers in parentheses refer to equations.

$\underset{\rightarrow}{A}$	Aerodynamic force vector applied to the vehicle not including thrust forces
$\underset{\rightarrow}{a}$	Vehicle centre-of-mass acceleration vector relative to inertial space
a	Speed of sound
b	Reference length (fuselage diameter for ballistic rocket vehicles)
C_D	(Total drag)/($\frac{1}{2}\varrho V^2 S$)
$C_{L_{fin}}$	(Fin lift normal to fin chord)/($\frac{1}{2}\varrho V^2 S$)
$(C_{L_\alpha})_{fin}$	$\left.\frac{\partial C_{L_{fin}}}{\partial \alpha}\right\rvert_e$
C_l	$L_{A_B}/(\frac{1}{2}\varrho V^2 Sb)$
C_{l_p}	$\left.\frac{2V}{b}\frac{\partial C_l}{\partial p_B}\right\rvert_e$
$C_{l_{\delta fin}}$	$\left.\frac{\partial C_l}{\partial \delta_{fin}}\right\rvert_e$
C_m	$M_{A_B}/(\frac{1}{2}\varrho V^2 Sb)$
C_{m_q}	$\left.\frac{2V}{b}\frac{\partial C_m}{\partial q_B}\right\rvert_e$
C_n	$N_{A_B}/(\frac{1}{2}\varrho V^2 Sb)$
C_{n_r}	$\left.\frac{2V}{b}\frac{\partial C_n}{\partial r_B}\right\rvert_e$

C_y	$Y_{A_B}/(\frac{1}{2}\rho V^2 S)$
$C_{y_{fin}}$	(Aerodynamic force along y-axis of F_B generated by pseudo fin)/($\frac{1}{2}\rho V^2 S$)
$C_{y_{ofin}}$	$C_{y_{fin}}$ for $\alpha_{fin} = 0$
$(C_{y\alpha})_{fin}$	$\left.\dfrac{\partial C_{y_{fin}}}{\partial \alpha_{fin}}\right\rvert_e$
$C^B_{y_\beta}$	$\left.\dfrac{\partial C^B_y}{\partial \beta}\right\rvert_e$
C^B_z	$Z_{A_B}/(\frac{1}{2}\rho V^2 S)$
$C^B_{z_{fin}}$	(Aerodynamic force along z-axis of F_B generated by pseudo fin)/($\frac{1}{2}\rho V^2 S$)
$C_{z_{ofin}}$	$C_{z_{fin}}$ for $\alpha_{fin} = 0$
$C_{z\alpha}$	$\left.\dfrac{\partial C_z}{\partial \alpha}\right\rvert_e$
$(C_{z\alpha})_{fin}$	$\left.\dfrac{\partial C_{z_{fin}}}{\partial \alpha_{fin}}\right\rvert_e$
D	Total drag
e_{fin}	The body-fin interference factor of the pseudo fin
$\underrightarrow{F}$	External force vector acting on the vehicle centre-of-mass
F_B	Body-fixed reference frame, see Figure 2
F_B'	Modified body-fixed reference frame, see Figure 2
F_I	Inertial reference frame, see Figure 1
F_L	Launcher inertial reference frame, see Figure 1
F_R	Structural body-fixed reference frame, see Figure 2

F_T	Stand-off inertial reference frame, see Figure 1
g	Acceleration due to gravity
g_o	Nominal sea level acceleration due to gravity
$\underrightarrow{h}$	Vehicle angular momentum vector about centre-of-mass
h_{ASL}	Altitude of vehicle centre-of-mass above sea level
I_{sp_i}	Specific impulse of i-th rocket motor
I^B_{xx}, I^B_{yy}, ...	Vehicle moments of inertia about its centre-of-mass written as components in F_B
$(L_{A_B}, M_{A_B}, N_{A_B})$	Aerodynamic moment components in F_B not including thrust moments, about centre-of-mass
$(L_{T_B}, M_{T_B}, N_{T_B})$	Thrust moment components in F_B about centre-of-mass
$(L_{T_B}, M_{T_B}, N_{T_B})_{cg}$	See equation (2.6.1)
$(L_{T_B}, M_{T_B}, N_{T_B})_{nz}$	See equation (2.6.1)
$\underrightarrow{M}_A$	Aerodynamic moment vector acting about the vehicle centre-of-mass not including thrust moments
$\underrightarrow{M}_T$	Thrust moment vector acting about the vehicle centre-of-mass
m	Vehicle total mass
m_{em}	Airframe mass
$(m_{Me})_i$	Mass of i-th rocket motor less propellant
M_{PL}	Payload mass
$(m_{PR})_i$	Mass of i-th rocket motor's propellant
N_M	Total number of motors
p_A	Atmospheric pressure
(p_B, q_B, r_B)	Angular velocity components of F_B with respect to F_I written as components in F_B

q_D	Dynamic presure, $\frac{1}{2}\rho V^2$
$\vec{R}$	Position vector of vehicle centre-of-mass relative to F_I
$\vec{R}_T$	Position vector of vehicle centre-of-mass relative to F_T
$\vec{R}_{TI}$	Position vector of R_T relative to F_I
R	See equation (2.9,5)
r_E	Radius of the Earth to the nominal sea level datum plane
S	Reference area (fuselage cross-sectional area for ballistic rocket vehicles)
s	Distance the vehicle has moved along the launcher from its initial position [see equation (2.7,3)]
s_G	Launch rail guide length (see Figure 5)
$\vec{T}$	Thrust vector
T_A	Atmospheric temperature
T_i	Thrust of the i-th rocket motor
(u_B, v_B, w_B)	Components of $\vec{V}$ in F_B
$(u_{B_E}, v_{B_E}, w_{B_E})$	Components of $\vec{V}_E$ in F_B
$(u_{B_g}, v_{B_g}, w_{B_g})$	Components of $\vec{W}$ in F_B
$\vec{V}$	Airspeed vector
$\vec{V}_E$	Velocity vector of the vehicle centre-of-mass with respect to F_I
V	Magnitude of $\vec{V}$
V_{xz}	$\sqrt{u_B^2 + w_B^2}$
$\vec{W}$	Wind velocity vector with respect to F_I
(W_1, W_2, W_3)	Components of $\vec{W}$ in F_I

w_{fin}	Pseudo fin airspeed component normal to its chord plane, see equation (2.4,11)
$(X_{A_B}, Y_{A_B}, Z_{A_B})$	Aerodynamic force components in F_B not including thrust contribution acting at the vehicle centre-of-mass
$(X_{T_B}, Y_{T_B}, Z_{T_B})$	Thrust vector components in F_B
(x_{ac}, y_{ac}, z_{ac})	Coordinates of the vehicle aerodynamic centre in F_R
$(x_{ac_{fin}}, y_{ac_{fin}}, z_{ac_{fin}})$	Coordinates of the aerodynamic centre of the pseudo fin in F_R
(x_{cg}, y_{cg}, z_{cg})	Coordinates of the vehicle centre-of-mass in F_R
(x_{em}, y_{em}, z_{em})	Coordinates of the airframe (empty) centre-of-mass in F_R
(x_T, y_T, z_T)	Components of $\underset{\rightarrow}{R}_T$ in F_T [see equation (2.10,3)]
(x_I, y_I, z_I)	Components of $\underset{\rightarrow}{R}$ in F_I
$[(x_{Me})_i, (y_{Me})_i, (z_{Me})_i]$	Coordinates of the centre-of-mass of the empty motor case of the i-th rocket motor
(x_{PL}, y_{PL}, z_{PL})	Coordinates of the payload centre-of-mass in F_R
$[(x_{PR})_i, (y_{PR})_i, (z_{PR})_i]$	Coordinates of the centre-of-mass of the propellant of the i-th rocket motor
α	Angle of attack of vehicle [see equation (2.4,6a)]
α_{fin}	Angle of attack of pseudo fin's chord plane [see equation (2.4,12b)]
β	Sideslip angle of vehicle [see equation (2.4,6b)]
δ_{fin}	Cant angle of pseudo fin (see Figure 4)
θ_B	Elevation Euler angle of F_B
ξ_A	Aspect elevation angle of vehicle relative to F_T
ξ_E	Aspect azimuth angle of vehicle relative to F_T

ϱ	Air density
θ_B	Euler bank angle for F_B
θ_{fin}	Cylindrical coordinate of pseudo fin (see Figure 4)
ψ_B	Euler azimuth angle for F_B
$\underrightarrow{\omega}_B$	Angular velocity vector of F_B with respect to F_I

NOTATION CONVENTIONS

F_A	Reference frame A
$\underrightarrow{X}$	Vector quantity X
$\underrightarrow{X} \times \underrightarrow{Y}$	Vector cross product of X and Y
$\underline{X}$	A matrix X
$\underline{X}^T$	The transpose of $\underline{X}$ or the components of a vector $\underrightarrow{X}$ expressed in the reference frame F_T (context will determine which interpretation is intended)
$\underline{X}_A$	$\underrightarrow{X}$ expressed as components in F_A
X'	A $F_B{}'$ variable that is analogous to the variable X in F_B
$\underline{x}$	A column matrix x
x_e	A quantity x whose value is computed for an aerodynamic quasisteady equilibrium condition

DEFENCE RESEARCH ESTABLISHMENT SUFFIELD
RALSTON ALBERTA

SUFFIELD MEMORANDUM NO. 1081

A NONLINEAR SIX DEGREE-OF-FREEDOM BALLISTIC AERIAL TARGET SIMULATION MODEL (U)

by

A.B. Markov

1. INTRODUCTION

In 1979 DRES began development work towards modifying the U.S. army BATS target for use with the higher specific impulse CRV7 rocket motors under the auspices of a joint US-Canadian TTCP agreement. This original work has lead to a number of DRES initiated activities including the modification of the target to permit multistaging and the development of an all CRV7, multistaged vehicle referred to as ROBOT-9 (Figure 1). As well, support equipment has been developed for target operation in moderately heavy seas, i.e. in conditions typical of Canadian coastal waters (the ROBOT System development history is summarized in more detail in Reference 1).

To support the development of these free-flight targets, computer simulation programs were required that predicted the dynamic characteristics of the vehicles. Of particular importance were accurate predictions of basic performance parameters (e.g. range and flight-time), wind effects, effect of nonstandard atmospheric conditions and the dynamic effects of launching from a moving ship on a finite, nonzero length launcher.

No existing software was available to DRES that performed all of these tasks while conveniently permitting some configuration variation. As a result, in the period September, 1980 to December, 1981, a six degree-of-freedom simulation package was written, debugged, and tested at DRES. This package was applied to the evaluation of the CRV7/BATS and ROBOT-9 performance characteristics and safety-envelopes. It was coded in IBM Fortran and has been used on the IBM 3033 computer with the H-extended compiler. The package is currently being installed on a VAX 11/780 computer for use with a FORTRAN 77 compiler, and will be adapted for use with the Honeywell DPS-8/70C computer. The package has been designated BALSIM

It is the intent of this report to provide documentation of BALSIM in sufficient detail to permit users familiar with FORTRAN to run the program. Chapter 2 develops the dynamic model and summarizes its limitations. Chapter 3 describes the BALSIM package in general terms. Finally, Appendix 1 is intended as an essentially self-contained userbook for the package, and includes a listing of all program modules.

2. DYNAMIC MODEL

This section summarizes the key features of the dynamic model programmed into the BALSIM package. The basic six degree-of-freedom equations are derived in the following sections.

2.1 Fundamental Assumptions

Several overall simplifying assumptions have been made in the derivation of the equations of motion. They are valid for ballistic rocket vehicles that have rigid structures and relatively short ranges, i.e. less than 100 km (50 nm).

The assumptions are as follows:

1. The Earth is flat and any Earth-fixed reference frame is inertial.
2. The vehicle is a rigid body.
3. There are no control surfaces.

Assumption 3 may be readily relaxed by adding the appropriate control terms into the equations of motion.

2.2 Reference Frames, Rotation Matrices and Angular Velocities

In the general case both an Earth-fixed (say F_E) and an inertial reference frame (say F_I) must be defined. Because of the first simplifying assumption of the previous section, these reference frames become identical. Only F_I will be used here. Thus let F_I be an Earth-fixed inertial reference frame whose origin is at the launch site, whose x-axis points along the projection of the nominal launch trajectory onto the Earth's surface and whose z-axis is nominally downwards (see Fig. 1). The y-axis follows from the right hand rule.

A second, inertial Earth-fixed reference frame that is useful is the launcher reference frame F_L. The origin of F_L is located at the launch site with the x-axis pointing in the launch direction and the z-axis being nominally downwards (see Fig. 1). The y-axis follows from the right hand rule.

A third Earth-fixed reference frame which is occasionally required is that of a reference frame F_T placed at some distance from the launch site. This reference frame may be used to compute the rocket aspect angle presentations (e.g. from the training ship). Since the location and orientation of this reference will depend on the particular application, it is defined only generally in Figure 1.

Since the aerodynamic forces are most conveniently expressed with respect to the vehicle, a body-fixed reference frame F_B will also be used. The origin of F_B is located at the vehicle centre-of-mass. In vehicles that are axisymmetric, the x-axis is on the axis of symmetry and points forward through the nose. Otherwise the x-axis points in the nominal launch direction. The z-axis is nominally downward, while the y-axis follows from the right hand rule. This reference frame and some associated aerodynamic angles are shown in Figure 2.

For cases where the rocket vehicle mass characteristics are axisymmetric, the aerodynamic forces are independent of the vehicle's roll attitude, and the thrust forces are axisymmetric, the body-fixed reference frame need not spin with the vehicle. Thus a reference frame F_B' is defined which is identical to F_B except that it does not rotate with the vehicle about the axis of symmetry. Initially F_B and F_B' will coincide.

A third body-fixed reference frame that is useful in specifying the vehicle's mass, inertia and configuration characteristics is a reference frame F_R whose origin is located on a nose datum plane on the vehicle. If the vehicle is axisymmetric, then the origin of F_R

is on the axis of symmetry, and its x-axis points towards the rear of the vhicle on the axis of symmetry. Otherwise, the origin of F_R may be any convenient location on the nose datum plane, and the x-axis nominally points towards the rear of the vehicle. The z-axis is nominally downward and the y-axis follows from the right hand rule (see Figure 2).

In the following, use is made of a number of notation conventions. In particular if $\underline{L}_{BA}$ denotes a rotation matrix relating the components of a vector $\underset{\rightarrow}{V}$ expressed in $F_A(\underline{v}_A)$ to the components of the same vector in $F_B(\underline{v}^B)$, then

$$\underline{v}^B = \underline{L}_{BA}\underline{v}^A \tag{2.2,1}$$

The following definitions, geometric relationships, and matrices, will be employed in the presentation of the equations of motion.

1. The rotation matrix relating F_I and F_B:

$$\underline{L}_{BI} = \begin{bmatrix} \cos\theta_B\cos\psi_B & \cos\theta_B\sin\psi_B & -\sin\theta_B \\ \begin{array}{r}\sin\phi_B\sin\theta_B\cos\psi_B \\ -\cos\phi_B\sin\psi_B\end{array} & \begin{array}{r}\sin\phi_B\sin\theta_B\sin\psi_B \\ +\cos\phi_B\cos\psi_B\end{array} & \sin\phi_B\cos\theta_B \\ \begin{array}{r}\cos\phi_B\sin\theta_B\cos\psi_B \\ +\sin\phi_B\sin\psi_B\end{array} & \begin{array}{r}\cos\phi_B\sin\theta_B\sin\psi_B \\ -\sin\phi_B\cos\psi_B\end{array} & \cos\phi_B\cos\theta_B \end{bmatrix} \tag{2.2,2a}$$

$$= [l_{BI_{ij}}] \tag{2.2,2b}$$

where ϕ_B, θ_B, and ψ_B are the Euler angles defined by Etkin (Reference 3).

The rotation matrix relating F_I and $F_B{}'$ follows from (2.2,2a) by substituting $\psi_B{}'$, $\theta_B{}'$ and $\phi_B{}'$ for ψ_B, θ_B and ϕ_B respectively, and will be denoted $L_B{}'_I$.

2. The rotation matrix relating F_L and F_I:

$$\underline{L}_{LI} = \begin{bmatrix} \cos\theta_{B_o} & 0 & -\sin\theta_{B_o} \\ 0 & 1 & 0 \\ \sin\theta_{B_o} & 0 & \cos\theta_{B_o} \end{bmatrix} \tag{2.2,3a}$$

$$= [l_{LI_{ij}}] \tag{2.2,3b}$$

3. The rotation matrix relating F_T and F_I:

$$\underline{L}_{TI} = \begin{bmatrix} \cos\psi_T & \sin\psi_T & 0 \\ -\sin\psi_T & \cos\psi_T & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.2,4a}$$

$$= [l_{TI_{ij}}] \tag{2.2,4b}$$

4. The angular velocity of the vehicle with respect to F_I written as components in F_B and F_B':

$$\underline{\omega}^B = (p_B, q_B, r_B)^T \tag{2.2,5a}$$

$$\underline{\omega}^{B'} = (0, q_B', r_B')^T \tag{2.2,5b}$$

5. The angular rate cross-product matrices (Reference 3) in F_B and F_B':

$$\underline{\tilde{\omega}}^B = \begin{bmatrix} 0 & -r_B & q_B \\ r_B & 0 & -p_B \\ -q_B & p_B & 0 \end{bmatrix} \tag{2.2,6a}$$

$$\underline{\tilde{\omega}}^{B'} = \begin{bmatrix} 0 & -r_B' & q_B' \\ r_B' & 0 & 0 \\ -q_B' & 0 & 0 \end{bmatrix} \tag{2.2,6b}$$

6. The airspeed vector of the vehicle written as components in F_B and F_B':

$$\underline{V}^B = (u_B, v_B, w_B)^T \tag{2.2,7a}$$

$$\underline{V}^{B'} = (u_B, v_B', w_B')^T \tag{2.2,7b}$$

7. The groundspeed vector of the vehicle with respect to F_I written as components in F_B, F_B' and F_I:

$$\underline{V}_E^B = (u_{B_E}, v_{B_E}, w_{B_E})^T \qquad (2.2,8a)$$

$$\underline{V}_E^{B'} = (u_{B_E}, v_{B_E}', w_{B_E}')^T \qquad (2.2,8b)$$

$$\underline{V}_E^I = (\dot{x}_I, \dot{y}_I, \dot{z}_I)^T \qquad (2.2,8c)$$

8. The aerodynamic angles (see Figure 2):

$$\alpha = \arctan (w_B/u_B) \qquad (2.2,9)$$

$$\beta = \arctan (v_B/V_{xz}) \qquad (2.2,10)$$

where

$$V_{xz} = \sqrt{u_B^2 + w_B^2} \qquad (2.2,11)$$

$$V = \sqrt{u_B^2 + v_B^2 + w_B^2} \qquad (2.2,12)$$

Here α is the angle of attack of the x-axis of F_B, β is the sideslip angle of the x-axis of F_B, V is the airspeed, and V_{xz} is the magnitude of the airspeed vector component along the x-z plane of F_B.

Analogous angles to α and β may be written in terms of F_B' components by direct substitution of F_B' quantities for F_B quantities.

9. The geometric relationships (see Figure 2):

$$u_B = V \cos\beta \cos\alpha \qquad (2.2,13a)$$

$$v_B = V \sin\beta \qquad (2.2,13b)$$

$$w_B = V \cos\beta \sin\alpha \qquad (2.2,13c)$$

10. The wind velocity with respect to F_I written as components in F_I, F_B, and F_B':

$$\underline{W}^I = (W_1, W_2, W_3)^T \qquad (2.2,14a)$$

$$\underline{W}^B = (u_{B_g}, v_{B_g}, w_{B_g})^T \qquad (2.2,14b)$$

$$\underline{W}^{B'} = (u_{B_g}, v_{B_g}', w_{B_g}')^T \qquad (2.2,14c)$$

11. The acceleration due to gravity written as components in F_I:

$$\underline{g}^I = (0, 0, g)^T \qquad (2.2,15)$$

12. The aerodynamic forces (not including thrust forces) written as components in F_B and F_B':

$$\underline{A}_A^B = (X_{A_B}, Y_{A_B}, Z_{A_B})^T \qquad (2.2,16a)$$

$$\underline{A}_A^{B'} = (X_{A_B}, Y_{A_B}', Y_{A_B}')^T \qquad (2.2,16b)$$

13. The aerodynamic moments (not including thrust moments) written as components in F_B and F_B':

$$\underline{M}_A^B = (L_{A_B}, M_{A_B}, N_{A_B})^T \qquad (2.2,17a)$$

$$\underline{M}_A^{B'} = (L_{A_B}, M_{A_B}', M_{A_B}')^T \qquad (2.2,17b)$$

14. The inertia matrix of the vehicle with respect to its centre-of-mass expressed in F_B (see Etkin, Reference 3) and F_B':

$$\underline{I}^B = \begin{bmatrix} I_{xx}^B & -I_{xy}^B & -I_{xz}^B \\ -I_{xy}^B & I_{yy}^B & -I_{yz}^B \\ -I_{xz}^B & -I_{yz}^B & I_{zz}^B \end{bmatrix} \qquad (2.2,18a)$$

$$\underline{I}^{B'} = \begin{bmatrix} I_{xx}^B & 0 & 0 \\ 0 & I_{yy}^{B'} & 0 \\ 0 & 0 & I_{yy}^{B'} \end{bmatrix} \qquad (2.2,18b)$$

15. The total thrust forces written as components in F_B and F_B':

$$\underline{T}^B = (X_{T_B}, Y_{T_B}, Z_{T_B})^T \qquad (2.2,19a)$$

$$\underline{T}^{B'} = (X_{T_B}, Y_{T_B}', Y_{T_B}')^T \qquad (2.2,19b)$$

16. The total thrust moments as components in F_B and F_B':

$$\underline{M}_T^B = (L_{T_B}, M_{T_B}, N_{T_B})^T \tag{2.2,20a}$$

$$\underline{M}_T^{B'} = (L_{T_B}, M_{T_B}', M_{T_B}')^T \tag{2.2,20b}$$

2.3 Newton-Euler Development of the General Equations of Motion

Newton-Euler techniques begin with the fundamental equations (Reference 3)

$$\vec{F} = m\vec{a} \tag{2.3,1}$$

and

$$\dot{\vec{h}} = \vec{M} \tag{2.3,2}$$

$\vec{a}$ is the acceleration vector of the body centre-of-mass relative to an inertial reference frame, $\vec{h}$ is the angular momentum of the body about its centre-of-mass, $\vec{F}$ is the external force vector acting at the centre-of-mass and $\vec{M}$ is the external moment vector about the centre-of-mass. $\vec{F}$ may be written

$$\vec{F} = m\dot{\vec{V}}_E \tag{2.3,3}$$

where $\vec{V}_E$ is the velocity vector of the vehicle with respect to F_I and m is mass of the vehicle. An expression for $\vec{h}$ follows from the fundamental relationship

$$\vec{h} = \int_{mass} [\vec{r} \times \dot{\vec{r}}]\, dm \tag{2.3,4a}$$

or

$$\vec{h} = \int_{mass} [\vec{r} \times \mathring{\vec{r}} + \vec{r} \times (\vec{\omega}_B \times \vec{r})]\, dm \tag{2.3,4b}$$

where $\vec{r}$ is the position vector of an element of mass dm of the body with respect to its centre-of-mass (see Figure 3), $\vec{\omega}_B$ is the angular velocity vector of F_B with respect to F_I, '.' when applied to a vector represents rate of change with respect to F_I and '∘' when applied to a vector represents rate of change with respect to F_B (see Reference 4 for a more thorough discussion of vector differentiation). Equation (2.3,4b) may be written in matrix notation as (replacing $\vec{\omega}_B \times \vec{r}$ by $-\vec{r} \times \vec{\omega}_B$ and dropping the subscript 'B' on $\vec{\omega}_B$ for the sake of brevity)*

* Superscripts on matrix quantities refer to the reference frame in which the components of the matrix are expressed. Overscore '~' refers to the matrix equivalent of the vector cross-product.

$$\underline{h}_B = \int_{mass} [\underline{\tilde{r}}^B \underline{\dot{r}}^B - \underline{\tilde{r}}^B \underline{\tilde{r}}^B \underline{\omega}^B]\, dm \quad (2.3,5)$$

But

$$\underline{\dot{r}}^B = \underline{0} \quad (2.3,6)$$

for a rigid body. Since $\underline{\omega}^B$ is a constant with respect to the integration in (2.3,5), it follows that

$$\underline{h}^B = \underline{I}^B \underline{\omega}^B \quad (2.3,7)$$

where

$$\underline{I}^B = -\int_{mass} \underline{\tilde{r}}^B \underline{\tilde{r}}^B\, dm \quad (2.3,8)$$

$\underline{I}^B$ is, by convention, given by (2.2,18a).

The externally applied force $\underset{\rightarrow}{F}$ is made up of an aerodynamic component $\underset{\rightarrow}{A}$, a thrust component $\underset{\rightarrow}{T}$ and a gravitational component $m\underset{\rightarrow}{g}$ such that

$$\underset{\rightarrow}{F} = \underset{\rightarrow}{A} + \underset{\rightarrow}{T} + m\underset{\rightarrow}{g} \quad (2.3,9)$$

Substituting (2.3,9) into (2.3,1), the vector force equation becomes

$$m\underset{\rightarrow}{V}_E = \underset{\rightarrow}{A} + \underset{\rightarrow}{T} + m\underset{\rightarrow}{g} \quad (2.3,10)$$

The externally applied moment $\underset{\rightarrow}{M}$ is made up of an aerodynamic component $\underset{\rightarrow}{M}_A$ and a thrust component $\underset{\rightarrow}{M}_T$ such that

$$\underset{\rightarrow}{M} = \underset{\rightarrow}{M}_A + \underset{\rightarrow}{M}_T \quad (2.3,11)$$

Substituting (2.3,11) into (2.3,2) yields

$$\underset{\rightarrow}{\dot{h}} = \underset{\rightarrow}{M}_A + \underset{\rightarrow}{M}_T \quad (2.3,12)$$

Other than the gravitational force, the dominant forces and moments acting on the aircraft are due to aerodynamic causes and are largely determined by its orientation and configuration. It is accordingly advantageous to write the matrix equations of motion with respect to a body-fixed reference frame. This reference frame is chosen to be F_B. Furthermore, this choice does not introduce any gravitational moments since the origin of F_B and the centre-of-mass of the vehicle coincide.

Thus the matrix force and moment equations become

$$m(\dot{\underline{V}}_E^B + \tilde{\underline{\omega}}^B \underline{V}_E^B) = \underline{A}^B + \underline{T}^B + m\underline{L}_{BI}\,\underline{g}^I \tag{2.3,13}$$

and

$$\dot{\underline{h}}^B + \tilde{\underline{\omega}}^B \underline{h}^B = \underline{M}_A^B + \underline{M}_T^B \tag{2.3,14}$$

Substituting for $\underline{h}_B$ from (2.3,7), the moment equation becomes

$$\underline{I}^B \dot{\underline{\omega}}^B + \dot{\underline{I}}^B \underline{\omega}^B + \tilde{\underline{\omega}}^B \underline{I}^B \underline{\omega}^B = \underline{M}_A^B + \underline{M}_T^B \tag{2.3,15}$$

Writing out the equation (2.3,13) in scalar form yields

$$m(\dot{u}_{B_E} + q_B w_{B_E} - r_B v_{B_E}) = X_{A_B} + X_{T_B} + mgl_{BI_{13}} \tag{2.3,16a}$$

$$m(\dot{v}_{B_E} + r_B u_{B_E} - p_B w_{B_E}) = Y_{A_B} + Y_{T_B} + mgl_{BI_{23}} \tag{2.3,16b}$$

$$m(\dot{w}_{B_E} + p_B v_B - q_B u_{B_E}) = Z_{A_B} + Z_{T_B} + mgl_{BI_{33}} \tag{2.3,16c}$$

for the force equations, and

$$\begin{aligned} I_{xx}^B \dot{p}_B - I_{xy}^B \dot{q}_B - I_{xz}^B \dot{r}_B + \dot{I}_{xx}^B p_B - \dot{I}_{xy}^B q_B - \dot{I}_{xz}^B r_B + I_{yz}(r_B^2 - q_B^2) \\ + (I_{zz}^B - I_{yy}^B) r_B q_B + I_{xy}^B r_B p_B - I_{xz}^B q_B p_B = L_{A_B} + L_{T_B} \end{aligned} \tag{3.2,17a}$$

$$\begin{aligned} - I_{xy}^B \dot{p}_B + I_{yy}^B \dot{q}_B - I_{yz}^B \dot{r}_B - \dot{I}_{xy}^B p_B + \dot{I}_{yy}^B q_B - \dot{I}_{yz}^B r_B + I_{xz}^B(p_B^2 - r_B^2) \\ + (I_{xx}^B - I_{zz}^B) p_B r_B + I_{yz}^B p_B q_B - I_{xy}^B r_B q_B = M_{A_B} + M_{T_B} \end{aligned} \tag{3.2,17b}$$

$$\begin{aligned} - I_{xz}^B \dot{p}_B - I_{yz}^B \dot{q}_B + I_{zz}^B \dot{r}_B - \dot{I}_{xz}^B p_B - \dot{I}_{yz}^B q_B + \dot{I}_{zz}^B r_B + I_{xy}^B(q_B^2 - p_B^2) \\ + (I_{yy}^B - I_{xx}^B) p_B q_B + I_{xz}^B q_B r_B - I_{yz}^B p_B r_B = N_{A_B} + N_{T_B} \end{aligned} \tag{3.2,17c}$$

for the moment equations.

Kinematic equations are also required for the linear and rotational position of the aircraft. The linear position equations follow from

$$\underline{V}_E^I = \underline{L}_{IB}\,\underline{V}_E^B \tag{2.3,18}$$

The rotational position equations are the Euler angle rate equations and are derived in Etkin (Reference 3). The resulting scalar kinematic equations of motion are thus seen to be

$$\dot{x}_I = l_{BI_{11}} u_{B_E} + l_{BI_{21}} v_{B_E} + l_{BI_{31}} w_{B_E} \tag{3.2,19a}$$

$$\dot{y}_I = l_{BI_{12}} u_{B_E} + l_{BI_{22}} v_{B_E} + l_{BI_{32}} w_{B_E} \tag{3.2,19b}$$

$$\dot{z}_I = l_{BI_{13}} u_{B_E} + l_{BI_{23}} v_{B_E} + l_{BI_{33}} w_{B_E} \tag{3.2,19c}$$

for linear position, and

$$\dot{\phi}_B = p_B + q_B \sin\phi_B \tan\theta_B + r_B \cos\phi_B \tan\theta_B \tag{2.3,20a}$$

$$\dot{\theta}_B = q_B \cos\phi_B - r_B \sin\phi_B \tag{2.3,20b}$$

$$\dot{\psi}_B = [q_B \sin\phi_B + r_B \cos\phi_B] \sec\theta_B \tag{2.3,20c}$$

for angular position.

It should be stressed that the variables (u_{B_E}, v_{B_E}, w_{B_E}) in equations (2.3,16), (2.3,17) and (2.3,19) are the body-axes components of the vehicle's ground velocity vector. This is not the same as the equations developed in Reference 2 where airspeed vector components in body-axes are used. Also, no assumptions have been made, up to this point, regarding vehicle planes of symmetry and the symmetry of the thrust and aerodynamic forces and moments. Finally, it should be noted that no assumptions have been made about the mass and inertia characteristics of the vehicle, i.e. in general

$$\dot{m} \neq 0$$

and

$$\dot{I}^B_{ij} \neq 0$$

This, too, is an added feature not present in the equations developed in Reference 2.

A number of simplifications may be added to these equations if certain symmetry conditions are satisfied. If the vehicle has mass symmetry about the xy and xz planes, then

$$I^B_{xz} = I^B_{xy} = I^B_{yz} = 0 \tag{2.3,21}$$

If the vehicle mass characteristics are also axisymmetric, then in addition to (2.3,21) we also have

$$I^B_{yy} = I^B_{zz} \tag{2.3,22}$$

Applying assumption (2.3,21) to the moment equations (2.3,17a) through to (2.3,17c) results in the simplified set of equations

$$\dot{p}_B = [-\dot{I}^B_{xx} p_B - (I^B_{zz} - I^B_{yy}) r_B q_B + L_{A_B} + L_{T_B}]/I^B_{xx} \tag{2.3,23a}$$

$$\dot{q}_B = [-\dot{I}^B_{yy} q_B - (I^B_{xx} - I^B_{zz}) p_B r_B + M_{A_B} + M_{T_B}]/I^B_{yy} \tag{2.3,23b}$$

$$\dot{r}_B = [-\dot{I}^B_{zz} r_B - (I^B_{yy} - I^B_{xx}) p_B q_B + N_{A_B} + N_{T_B}]/I^B_{zz} \tag{2.3,23c}$$

Applying the axisymmetry assumption (2.3,22) to those equations simplifies (2.3,23a) even further by eliminating the $(I^B_{zz} - I^B_{yy}) r_B q_B$ term, i.e.

$$\dot{p}_B = [-\dot{I}^B_{xx} p_B + L_{A_B} + L_{T_B}]/I^B_{xx} \tag{2.3,24}$$

If we now make the assumptions that the thrust forces are axisymmetric, that the vehicle mass characteristics are axisymmetric, and that the vehicle aerodynamic characteristics are independent of the vehicle's roll orientation, then we may take advantage of the simplifications that will result to the equations of motion by expressing them in the reference frame F_B' rather than in F_B. Recall that in Section 2.2 we defined F_B' as being identical to F_B except that it does not rotate with the vehicle about the axis of symmetry. As a result, if

$$\underset{\rightarrow}{\omega}_B' = p_B' \underset{\rightarrow}{i}_B + q_B' \underset{\rightarrow}{j}_B + r_B' \underset{\rightarrow}{k}_B \tag{2.3,25}$$

is the angular velocity vector of F_B' relative to the inertial reference frame F_I expressed

as components in F_B', then the latter condition implies that

$$p_B' = 0 \tag{2.3,26}$$

for all t.

Also in general we will have

$$q_B' \neq q_B \tag{2.3,27a}$$

$$r_B' \neq r_B \tag{2.3,27b}$$

For the purpose of generating the aerodynamic forces, however, q_B' and r_B' may be treated as being interchangeable with q_B and r_B respectively because of the assumption that the aerodynamic forces are independent of the roll orientation. However, the resulting aerodynamic forces are now expressed in F_B' axes rather than in F_B axes.

There are also aerodynamic forces that are generated due to the vehicle's rolling angular velocity p_B. The latter is not identical to p_B', and thus an equation of motion for p_B will still have to be retained.

Formally the equations of motion in F_B' may be obtained from the equations of motion in F_B by setting $p_B = 0$ in all equations except the p_B equation, replacing $(u_{B_E}, v_{B_E}, w_{B_E})$ with $(u_{B_E}, v'_{B_E}, w'_{B_E})$, (q_B, r_B) with (q_B', r_B'), $(\phi_B, \theta_B, \psi_B)$ with $(\phi_B', \theta_B', \psi_B')$, (M_{A_B}, N_{A_B}) with (M'_{A_B}, M'_{A_B}), (M_{T_B}, N_{T_B}) with (M'_{T_B}, M'_{T_B}), $(X_{A_B}, Y_{A_B}, Z_{A_B})$ with $(X_{A_B}, Y'_{A_B}, Y'_{A_B})$, $(X_{T_B}, Y_{T_B}, Z_{T_B})$ with $(X_{T_B}, Y'_{T_B}, Y'_{T_B})$, and applying the assumptions discussed previously. Finally, the p_B equation from the F_B equations is retained.

The resulting equations of motion are as follows:

$$m(\dot{u}_{B_E} + q_B' w'_{B_E} - r_B' v'_{B_E}) = X_{A_B} + X_{T_B} + mgl'_{BI_{13}} \tag{2.3,28a}$$

$$m(\dot{v}'_{B_E} + r_B' u_{B_E}) = Y'_{A_B} + Y'_{T_B} + mgl'_{BI_{23}} \tag{2.3,28b}$$

$$m(\dot{w}'_{B_E} - q_B u_{B_E}) = Y'_{A_B} + Y'_{T_B} + mgl'_{BI_{33}} \tag{2.3,28c}$$

$$\dot{p}_B = [-\dot{I}^B_{xx} p_B + L_{A_B} + L_{T_B}]/I^B_{xx} \qquad (2.3,29a)$$

$$\dot{q}'_B = [-\dot{I}^B_{yy} q'_B + M'_{A_B} + M'_{T_B}]/I^B_{yy} \qquad (2.3,29b)$$

$$\dot{r}'_B = [-\dot{I}^B_{yy} r'_B + M'_{A_B} + M'_{T_B}]/I^B_{yy} \qquad (2.3,29c)$$

$$\dot{x}_I = l'_{BI_{11}} u_{B_E} + l'_{BI_{21}} v'_{B_E} + l'_{BI_{31}} w'_{B_E} \qquad (3.2,30a)$$

$$\dot{y}_I = l'_{BI_{12}} u_{B_E} + l'_{BI_{22}} v'_{B_E} + l'_{BI_{32}} w'_{B_E} \qquad (3.2,30b)$$

$$\dot{z}_I = l'_{BI_{13}} u_{B_E} + l'_{BI_{23}} v'_{B_E} + l'_{BI_{33}} w'_{B_E} \qquad (3.2,30c)$$

$$\dot{\phi}'_B = q'_B \sin\phi'_B \tan\theta'_B + r'_B \cos\phi'_B \tan\theta'_B \qquad (2.3,31a)$$

$$\theta'_B = q'_B \cos\phi'_B - r'_B \sin\phi'_B \qquad (2.3,31b)$$

$$\psi'_B = [q'_B \sin\phi'_B + r'_B \cos\phi'_B] \sec\theta_B \qquad (2.3,31c)$$

2.4 Aerodynamic Model

The equations of motion developed in the previous section contain terms (e.g. X_B) that represent the aerodynamic forces acting on the vehicle. In this section these terms are defined as functions of the vehicle's state.

Although more sophisticated techniques are available (see the discussion in Reference 2), for the purposes of rigid body six degree-of-freedom simulation, it is usually quite adequate to use a quasisteady aerodynamic model based on Bryan's aerodynamic derivative technique.

No attempt has been made here to generalize this model so that it applies equally well to all types of air vehicles. Rather, its form has been simplified so that it is suitable for use only with free-flight, ballistic, rocket-boosted vehicles.

The resulting model expressed as aerodynamic force and moment components in F_B is summarized below. No attempt is made to rationalize this model other than to state that its use has resulted in predicted trajectories that are in good agreement with measured flight characteristics (see, e.g., Reference 1) of CRV7/BATS and ROBOT-9 vehicles.

The aerodynamic forces are specified by

$$X_{A_B} = -C_D q_D S \tag{2.4,1a}$$

$$Y_{A_B} = C_{y\beta} \beta q_D S + q_D S(C_{y\alpha fin} + C_{y_{ofin}})_{pseudo} \tag{2.4,1b}$$

$$Z_{A_B} = C_{z\alpha} \alpha q_D S + q_D S(C_{z\alpha fin} + C_{z_{ofin}})_{pseudo} \tag{2.4,1c}$$

The various quantities in these equations are defined in the notation. It is important to note that q_D is the dynamic pressure given by

$$q_D = \tfrac{1}{2} \varrho V^2 \tag{2.4,2}$$

where ϱ is the air density and V is the airspeed given by

$$V = (u_B^2 + v_B^2 + w_B^2)^{1/2} \tag{2.4,3}$$

Here (u_B, v_B, w_B) are the airspeed vector components expressed in F_B, and in the presence of nonzero wind conditions will not be the same as $(u_{B_E}, v_{B_E}, w_{B_E})$. Rather, they will be related to the wind velocity vector components in F_B $[(u_{B_g}, v_{B_g}, w_{B_g})]$ through the relationships

$$u_B = u_{B_E} - u_{B_g} \tag{2.4,4a}$$

$$v_B = v_{B_E} - v_{B_g} \tag{2.4,4b}$$

$$w_B = w_{B_E} - w_{B_g} \tag{2.4,4c}$$

The latter may be obtained by considering the fundamental vector relationship

$$\underset{\rightarrow}{V}_E = \underset{\rightarrow}{V} + \underset{\rightarrow}{W} \tag{2.4,5}$$

i.e. the ground velocity vector $\underset{\rightarrow}{V}_E$ equals the airspeed vector $\underset{\rightarrow}{V}$ plus the wind velocity vector $\underset{\rightarrow}{W}$.

C_D is the nondimensional drag coefficient, $C_{y\beta}$ is the aerodynamic derivative relating y-force due to sideslip angle β, $C_{z\alpha}$ is the aerodynamic derivative relating z-force due to angle of attack α, and S is a reference area that is usually the fuselage cross-sectional area for ballistic vehicles. C_D, $C_{y\beta}$, and $C_{z\alpha}$ may, in general, be Mach number and Reynolds number dependent, although here they are considered to be only Mach number dependent. α and β are given with respect to the x-axis of F_B, and from geometric considerations may be shown to be (see Figure 2)

$$\alpha = \arctan(w_B/u_B) \tag{2.4,6a}$$

$$\beta = \arctan(v_B/V_{xz}) \tag{2.4,6b}$$

where

$$V_{xz} = (u_B^2 + w_B^2)^{1/2} \tag{2.4,7}$$

Differential equations may be obtained for α and β by differentiating (2.4,6a) and (2.4,6b) with respect to time, with the results

$$\dot{\alpha} = [\dot{w}_B/V - \dot{u}_B\dot{w}_B/V^2]\cos^2\alpha \tag{2.4,8a}$$

and

$$\dot{\beta} = [V_{xz}\dot{v}_B - v_B(u_B\dot{u}_B + w_B\dot{w}_B)V_{xz}]/(V_{xz}^2\cos\beta) \tag{2.4,8b}$$

The last terms on the right hand sides of (2.4,1b) and (2.4,1c) are pseudo fin terms and are included to permit modeling of aerodynamic asymmetries (e.g. due to production tolerances). They do not include any of the effects produced by the vehicle's nominal fin configuration. The latter have already been included in C_D, $C_{y\beta}$ and $C_{z\alpha}$.

The pseudo fin terms are defined as follows:

$$C_{y_{ofin}} = -\, C_{L_{\alpha fin}}\delta_{fin}\sin\phi_{fin} \tag{2.4,9a}$$

$$C_{z_{ofin}} = -\, C_{L_{\alpha fin}}\delta_{fin}\cos\phi_{fin} \tag{2.4,9b}$$

$$C_{y\alpha fin} = -\, e_{fin}C_{L_{\alpha fin}}\sin\phi_{fin} \tag{2.4,10a}$$

$$C_{z\alpha fin} = -\, e_{fin}C_{L_{\alpha fin}}\cos\phi_{fin} \tag{2.4,10b}$$

$$w_{fin} = w_B \cos\phi_{fin} + v_B \sin\phi_{fin} \tag{2.4,11}$$

$$\alpha_{fin} = \arctan(w_{fin}/u_B) \tag{2.4,12}$$

Here $C_{L_{\alpha fin}}$ is the lift slope of the fin, δ_{fin} is the cant angle of the fin, ϕ_{fin} is the angular cylindrical coordinate of the fin, e_{fin} is a body-fin interference factor and α_{fin} is the angle of attack of the fin. The fin geometry coordinate system is summarized in Figure 4.

The aerodynamic moment expressions are defined similarly to the aerodynamic force expressions, as follows:

$$\begin{aligned} L_{A_B} = {} & C_{l_p} p_B q_D S b^2/(2V) + C_{l\delta fin} \delta_{fin} q_D S b \\ & - Y_{A_{B nom}} z_{cg} - Z_{A_{B nom}} y_{cg} \end{aligned} \tag{2.4,13a}$$

$$\begin{aligned} M_{A_B} = {} & Z_{A_{B nom}}(x_{ac} - x_{cg}) + C_{m_q} q_B S q_D b^2/(2V) \\ & + X_{A_B} z_{cg} + q_D S(C_{z\alpha fin}\alpha_{fin} + C_{z_{ofin}})_{pseudo}(x_{ac_{fin}} - x_{cg}) \end{aligned} \tag{2.4,13b}$$

$$\begin{aligned} N_{A_B} = {} & - Y_{A_{B nom}}(x_{ac} - x_{cg}) + C_{n_r} r_B S q_D b^2/(2V) \\ & + X_{A_B} y_{cg} - q_D S(C_{y\alpha fin} + C_{y_{ofin}})_{pseudo}(x_{ac_{fin}} - x_{cg}) \end{aligned} \tag{2.4,13c}$$

In these expressions $Y_{A_{B nom}}$ and $Z_{A_{B nom}}$ are given by (2.4,1b) and (2.4,1c) without the pseudo fin contributions, (x_{cg}, y_{cg}, z_{cg}) are the coordinates of the vehicle centre-of-mass in the vehicle structural reference frame F_R (see Figure 2), (x_{ac}, y_{ac}, z_{ac}) are the coordinates of the vehicle aerodynamic centre in F_R, $(x_{ac_{fin}}, y_{ac_{fin}}, z_{ac_{fin}})$ are the coordinates of the aerodynamic centre of the pseudo fin in F_R, b is the reference length (usually the fuselage diameter for ballistic vehicles) and the aerodynamic derivatives C_{l_p}, $C_{l\delta fin}$, C_{m_q}, $C_{z\alpha fin}$, C_{n_r}, $C_{y\alpha fin}$ are defined in the notation list.

The aerodynamic forces and moments may also be written as components in F_B by making an identical set of substitutions into (2.4,1) and (2.4,13) as used in converting the equations of motion written in F_B to those written in F_B'. It is important to note that certain simplifications result because of the underlying assumptions used in developing

the F_B' equations, i.e. no pseudo fin terms may be included and y_B' and z_B' axes characteristics are identical.

For the sake of brevity, the aerodynamic force and moment expressions in F_B' will not be given here.

2.5 Mass and Moments of Inertia Models

The equations of motion have been written so that variations in the vehicle's mass and inertia characteristics (due to rocket motor propellant burn) are permitted. Component methods are used to compute the total vehicle mass and moments of inertia. The components considered are the vehicle airframe, the vehicle payload, the vehicle rocket motors less propellant and the rocket motors' propellant. Of these components, only the propellant characteristics are considered to be variable with time. Finally, the assumption has been made that the payload, the rocket motors, and the propellant are point masses.

Under these conditions the expressions for the vehicle mass and inertia characteristics are summarized below. These equations are given for the reference frame F_B. Position coordinates are with respect to the vehicle structural reference frame F_R. The subscripts used to reference the different components are as follows:

1) 'em' — airframe (empty)
2) 'PL' — payload
3) 'Me' — rocket motors less propellant
4) 'PR' — rocket motor propellant

N_M is the total number of rocket motors. Other variables used are defined precisely in the notation.

The expressions for the mass and inertia characteristics are given by ($\Delta x_\xi = x_\xi - x_{cg}$, $\Delta y_\xi = y_\xi - y_{cg}$, and so forth)

$$m = m_{em} + m_{PL} + \sum_{i=1}^{N_M} [(m_{M_e})_i + (m_{PR})_i] \qquad (2.5,1)$$

$$I^B_{xx} = I^B_{xx_{em}} + m_{em}(\Delta y^2_{em} + \Delta z^2_{em}) + m_{PL}(\Delta y^2_{PL} + \Delta z^2_{PL}) + \sum_{i=1}^{N_M} \{(m_{M_e})_i[(\Delta y_{M_e})^2_i + (\Delta z_{M_e})^2_i] + (m_{PR})_i[(\Delta y_{PR})^2_i + (\Delta z_{PR})^2_i]\} \tag{2.5,2a}$$

$$I^B_{yy} = I^B_{yy_{em}} + m_{em}(\Delta x^2_{em} + \Delta z^2_{em}) + m_{PL}(\Delta x^2_{PL} + \Delta z^2_{PL}) + \sum_{i=1}^{N_M} \{(m_{M_e})_i[(\Delta x_{M_e})^2_i + (\Delta z_{M_e})^2_i] + (m_{PR})_i[(\Delta x_{PR})^2_i + (\Delta z_{PR})^2_i]\} \tag{2.5,2b}$$

$$I^B_{zz} = I^B_{zz_{em}} + m_{em}(\Delta x^2_{em} + \Delta y^2_{em}) + m_{PL}(\Delta x^2_{PL} + \Delta y^2_{PL}) + \sum_{i=1}^{N_M} \{(m_{M_e})_i[(\Delta x_{M_e})^2_i + (\Delta y_{M_e})^2_i] + (m_{PR})_i[(\Delta x_{PR})^2_i + (\Delta y_{PR})^2_i]\} \tag{2.5,2c}$$

$$I^B_{xy} = - m_{em}\Delta x_{em}\Delta y_{em} - m_{PL}\Delta x_{PL}\Delta y_{PL} - \sum_{i=1}^{N_M} \{(m_{M_e})_i(\Delta x_{M_e})_i(\Delta y_{M_e})_i + (m_{PR})_i(\Delta x_{PR})_i(\Delta y_{PR})_i\} \tag{2.5,2d}$$

$$I^B_{xz} = - m_{em}\Delta x_{em}\Delta z_{em} - m_{PL}\Delta x_{PL}\Delta z_{PL} - \sum_{i=1}^{N_M} \{(m_{M_e})_i(\Delta x_{M_e})_i(\Delta z_{M_e})_i + (m_{PR})_i(\Delta x_{PR})_i(\Delta z_{PR})_i\} \tag{2.5,2e}$$

$$I^B_{yz} = - m_{em}\Delta y_{em}\Delta z_{em} - m_{PL}\Delta y_{PL}\Delta z_{PL} - \sum_{i=1}^{N_M} \{(m_{M_e})_i(\Delta y_{M_e})_i(\Delta z_{M_e})_i + (m_{PR})_i(\Delta y_{PR})_i(\Delta z_{PR})_i\} \tag{2.5,2f}$$

$$x_{cg} = \{m_{em}x_{em} + m_{PL}x_{PL} + \sum_{i=1}^{N_M} [(m_{me})_i(x_{me})_i + (m_{PR})_i(x_{PR})_i]\}/m \tag{2.5,3a}$$

$$y_{cg} = \{m_{em}y_{em} + m_{PL}y_{PL} + \sum_{i=1}^{N_M} [(m_{me})_i(y_{me})_i + (m_{PR})_i(y_{PR})_i]\}/m \tag{2.5,3b}$$

$$z_{cg} = \{m_{em}y_{em} + m_{PL}y_{PL} + \sum_{i=1}^{N_M} [(m_{me})_i(z_{me})_i + (m_{PR})_i(z_{PR})_i]\}/m \tag{2.5,3c}$$

Because it has been assumed that the only mass changes are due to propellant burn, in these expressions the only time variable quantities will be $(m_{PR})_i$, $(x_{PR_i}, y_{PR_i}, z_{PR_i})$. If we further assume that the propellant burns in such a way that the centre-of-mass of the propellant of a given rocket motor does not change significantly (e.g. as would be the case in rocket motors that are **not** end burners), then $(x_{PR_i}, y_{PR_i}, z_{PR_i})$ are not time variable and only $(\dot{m}_{PR})_i$ need be considered. The latter is related to the specific impulse of the rocket motor through the relationship (Reference 5)

$$(\dot{m}_{PR})_i = T_i(t)/(I_{sp_i} g) \tag{2.5,4}$$

where g is the acceleration due to gravity, $T_i(t)$ is the thrust of the i-th rocket motor as a function of time t, and I_{sp_i} is the specific impulse of the i-th motor.

Under these assumptions and with (2.5,4), $\dot{m}$, the moment of inertia time derivatives, and $(\dot{x}_{cg}, \dot{y}_{cg}, \dot{z}_{cg})$ may be readily computed. For the sake of brevity, an exhaustive set of equations will not be given. Typically we have

$$\dot{m} = \sum_{i=1}^{N_M} [-T_i(t)/(I_{sp_i} g)] \tag{2.5,5}$$

$$\begin{aligned} \dot{I}^B_{xx} = & -2m_{em}(\Delta y_{em}\dot{y}_{cg} + \Delta z_{em}\dot{z}_{cg}) - 2m_{PL}(\Delta y\dot{y}_{cg} + \Delta z_{PL}\dot{z}_{cg}) \\ & + \sum_{i=1}^{N_M} \{-2(m_{M_e})_i[(\Delta y_{M_e})_i\dot{y}_{cg} + (\Delta z_{M_e})_i\dot{z}_{cg}] + (\dot{m}_{PR})_i \\ & [(\Delta y_{PR})_i^2 + (\Delta z_{PR})_i^2] - 2(m_{PR})_i[(\Delta y_{PR})_i\dot{y}_{cg} + (\Delta z_{PR})_i\dot{z}_{cg}]\} \end{aligned} \tag{2.5,6a}$$

$$\begin{aligned} \dot{I}^B_{xy} = & \, m_{em}(\dot{x}_{cg}\Delta y_{em} + \Delta x_{em}\dot{y}_{cg}) + m_{PL}(\dot{x}_{cg}\Delta y_{PL} + \dot{y}_{cg}\Delta x_{PL}) \\ & - \sum_{i=1}^{N_M} \{-(m_{M_e})_i[\dot{x}_{cg}(\Delta y_{M_e})_i + \dot{y}_{cg}(\Delta x_{M_e})_i] + (\dot{m}_{PR})_i \\ & (\Delta x_{PR})_i(\Delta y_{PR})_i - (m_{PR})_i[\dot{x}_{cg}(\Delta y_{PR})_i + \dot{y}_{cg}(\Delta x_{PR})_i]\} \end{aligned} \tag{2.5,6b}$$

$$\dot{x}_{cg} = \{\sum_{i=1}^{N_M} [-T_i(t)/(I_{sp_i})g](x_{PR})_i - x_{cg}\dot{m}\} m^{-1} \tag{2.5,7}$$

The mass and moment of inertia characteristic formulations expressed in the nonrotating reference frame F_B' follow from the F_B expressions by incorporating the simplifying assumptions used for writing the F_B' equations of motion (see Section 2.3). In particular, we have

$$I^B_{xy} = I^B_{xz} = I^B_{yz} = 0 \qquad (2.5,8)$$

and

$$I^B_{yy} = I^B_{zz} \qquad (2.5,9)$$

Equations (2.5,8) and (2.5,9) are just the result of the mass and inertia axisymmetry assumption used in developing the F_B' equations of motion.

For the sake of brevity, the expressions for the F_B' mass and inertia characteristics will not be given explicitly.

2.6 Thrust Characteristics

In Section 2.3 the equations of motion were written in the reference F_B with the thrust forces and moments written generally as $(X_{T_B}, Y_{T_B}, Z_{T_B})$ and $(L_{T_B}, M_{T_B}, N_{T_B})$ respectively. In this section these terms are examined in more detail.

The force terms $(X_{T_B}, Y_{T_B}, Z_{T_B})$ depend on the time domain thrust characteristics and the physical location and orientation of the rocket motors. This data must be known *a priori* to the simulation and is provided as input data to the computer program in the form of the thrust versus time look-up tables. The transformations used are summarized in Appendix 2.

The thrust moments require a somewhat more detailed examination. They are considered to consist of two components:

1) A moment due to the location and orientation of the thrust vector relative to F_B (see Figure 4),
2) A moment induced due to fixed vanes or nozzle grooves onto which the exhaust jet impinges.

Thus we have

$$L_{T_B} = (L_{T_B})_{cg} + (L_{T_B})_{nz} \qquad (2.6,1a)$$

$$M_{T_B} = (M_{T_B})_{cg} + (M_{T_B})_{nz} \qquad (2.6,1b)$$

$$N_{T_B} = (N_{T_B})_{cg} + (N_{T_B})_{nz} \qquad (2.6,1c)$$

These characteristics are rocket motor specific. It is assumed that such data is available for the rocket motors used in the simulation. It then follows that once the orientation and location of the rocket motor thrust vectors relative to the vehicle are specified, enough information is available to determine (L_{T_B}, M_{T_B}, N_{T_B}) as given by (2.6,1a) to (2.6,1c) (a detailed treatment is given in Appendix 2).

An assumption that has tacitly been made in this description of the thrust effects is that Coriolis forces and moments on the vehicle generated by the rocket motor exhaust are negligible. This need not always be the case, particularly for the moments, if the exhaust mass flow rate $\dot{m}$ and the exhaust velocity vector relative to the vehicle $\underset{\rightarrow}{V}_R$ are large. However, for vehicles in the class of CRV7/BATS and ROBOT-9 using short burn duration 70 mm (2.75 inch) rocket motors, these effects are negligible and will not be considered further in this report.

2.7 Vehicle Kinematic Restrictions While on Launcher

The presence of the launcher during the initial portion of the flight places a number of kinematic constraints on the vehicle's motion. This section considers these constraints for a rail launcher such as was used for CRV7/BATS and ROBOT-9 (see Reference 1).

The basic geometrical quantities are defined in Figure 5. The equations of motion while the vehicle is on the rail are presented for the following assumptions:

1) The vehicle is mechanically constrained from tipping backwards or forwards by the guide T-bolt until the T-bolt clears the launch rail (i.e. the vehicle is initially constrained to move along the x-axis of reference frame F_L). In the case of CRV7/BATS and ROBOT-9 there is also the launcher cage constraining the vehicle for part of its travel on the launch rail (see Reference 1).

2) The quantity s_G represents the distance the vehicle must move in the x-direction of F_B before the guide T-bolt clears the launch rail. The

bolt is assumed to be back far enough on the vehicle so that no significant tip-off may occur after it is clear of the rail and prior to the whole vehicle coming clear.

3) The vehicle may not move backwards on the launcher rail (i.e. $\dot{u}_B$ is never less than zero).

Under these assumptions it follows that if the distance that the vehicle centre-of-mass has travelled (s) is less than or equal to s_G, then the vehicle is physically constrained to move only in the launch rail direction, i.e. for $s \leqslant s_G$ we have

$$\dot{p}_B = \dot{q}_B = \dot{r}_B = \dot{v}_B = \dot{w}_B = 0 \qquad (2.7,1a)$$

$$\dot{u}_B \geqslant 0 \qquad (2.7,1b)$$

$$v_B = v_B(0) \qquad (2.7,2a)$$

$$w_B = w_B(0) \qquad (2.7,2b)$$

$$p_B = p_B(0) \qquad (2.7,2c)$$

$$q_B = q_B(0) \qquad (2.7,2d)$$

$$r_B = r_B(0) \qquad (2.7,2e)$$

The nonzero conditions (2.7,2a) to (2.7,2e) allow for a nonstationary launcher, i.e. as would be the case for a launch from a ship in linear and angular motion.

The quantity s is defined precisely as

$$s \triangleq \sqrt{(x_I - x_{I_0})^2 + (y_I - y_{I_0})^2 + z_I^2} \qquad (2.7,3)$$

where (x_I, y_I, z_I) are the vehicle centre-of-mass coordinates in F_I and $(x_{I_0}, y_{I_0}, 0)$ are the centre-of-mass coordinates when the vehicle is at rest on the launcher prior to first stage ignition.

For $s > s_G$, the governing equations are the unconstrained equations of motion developed in Section 2.3.

2.8 Wind Model

The aerodynamic model presented in Section 2.4 includes the wind velocities in F_B (F_B') as $(u_{B_g}, v_{B_g}, w_{B_g})$ $[(u_{B_g}, v'_{B_g}, w'_{B_g})]$, and has tacitly assumed that there is no variation of the wind velocity from one point to another. This is equivalent to assuming that the wind induced aerodynamic loads are determined by its velocity rate of change acting at the centre-of-mass of the vehicle, an assumption referred to as the uniform-gust approximation (References 2 and 3). This approximation is equivalent to assuming that the wind velocity spectral content significantly affecting the vehicle response is at wavelengths that are greater than the significant vehicle dimensions (Reference 3). This assumption is reasonable when considering the rigid body dynamic response of flight vehicles, particularly for smaller vehicles such as ROBOT-9 and CRV7/BATS.

The wind velocity vector components relative to F_I are most conveniently expressed as components in F_I, i.e. (W_1, W_2, W_3). These components may then be related to the wind velocity components in F_B with the rotation matrix $\underline{L}_{BI}$ as given by (2.2,2a), i.e.

$$(u_{B_g}, v_{B_g}, w_{B_g})^T = \underline{L}_{BI}(W_1, W_2, W_3)^T \qquad (2.8,1)$$

The simulation package has provisions for inputting (W_1, W_2, W_3) as functions of altitude. This allows modeling of wind velocity atmospheric boundary layer effects, vehicle encounters with jetstream regions, and so forth. As well, since meteorological winds aloft data is usually given as a function of altitude, simulation of measured wind conditions is facilitated.

2.9 Atmospheric Conditions

Since the ROBOT-9 and CRV7/BATS vehicles have the capability to achieve altitudes well above 9000 m (30,000 ft), an atmospheric model is required that takes into account variations in density (ϱ), temperature (T_A), pressure (p_A), and the speed of sound (a) as a function of altitude above sea level (h_{ASL}).

The models used are based on the U.S. standard atmosphere (1962), as is common practice in aeronautical engineering, and are valid within the troposphere, i.e. for $h_{ASL} \leqslant 11{,}100$ m (36,000 ft) (see Reference 6). They are given by

$$T_A = 288.15 - 0.0065\, h_{ASL}\, R \tag{2.9,1}$$

$$P_A = 101300.(T_A/288.15)^{5.255} \tag{2.9,2}$$

$$a = 20.0463\sqrt{T_A} \tag{2.9,3}$$

$$\varrho = 0.00348454\, p_A/T_A \tag{2.9,4}$$

where

$$R = r_E/(h_{ASL} + r_E) \tag{2.9,5}$$

T_A is in degrees Kelvin, p_A is in Pascals, a is in meters per second, ϱ is in kilograms per meter cubed, h_{ASL} is in meters, and r_E is the Earth's radius to the sea level datum,

$$r_e = 6.3567658 \times 10^6 \text{ m } (2.0855531 \times 10^7 \text{ ft}).$$

From the factor R it is also convenient to compute the variation of the acceleration due to gravity as a function of altitude, i.e.

$$g = g_o R^2 \tag{2.9,6}$$

where $g_o = 9.80667$ m/s² (32.1741 f/s²).

Provision has been made in the simulation package to vary the temperature and pressure (and thus the density) from the standard values by allowing altitude dependent per cent deviations from standard conditions.

2.10 Aspect Angle Equations

For target and flight test applications, it is frequently necessary that the vehicle's aspect azimuth (ξ_A) and elevation (ξ_E) angles be known with respect to an observer at F_T (see Figure 1). This section presents equations for ξ_A and ξ_E in terms of the location and orientation of F_T relative to that of F_I.

It is assumed that the x-y planes of F_I and F_T are parallel, i.e. that F_I may be rotated to F_T through a rotation ψ_T about the z-axis of F_I.

Let the vector position of F_T relative to F_I be $\underset{\rightarrow}{R}_{TI}$, and that of the vehicle centre-of-mass relative to the origin of F_I be $\underset{\rightarrow}{R}$ (see Figure 6). It follows that the vector position of the vehicle relative to F_T is given by

$$\underset{\rightarrow}{R}_T = \underset{\rightarrow}{R} - \underset{\rightarrow}{R}_{TI} \qquad (2.10,1)$$

or in matrix notation

$$\underline{R}^T_T = \underline{L}_{TI}\underline{R}^I - \underline{L}_{TI}\underline{R}^I_{TI} \qquad (2.10,2)$$

where $\underline{L}_{TI}$ is the rotation matrix rotating vector components in F_I to components in F_T, and is given by (2.2,4a).

Equations (2.10,2) may be written in scalar form as

$$x_T = (x_I - x_{TI})\cos\psi_T + (y_I - y_{TI})\sin\psi_T \qquad (2.10,3a)$$

$$y_T = -(x_I - x_{TI})\sin\psi_T + (y_I - y_{TI})\cos\psi_T \qquad (2.10,3b)$$

$$z_T = z_I - z_{TI} \qquad (2.10,3c)$$

From the definition of ξ_A and ξ_E in Figure 1, it follows that

$$\xi_A = \arctan(y_T/x_T) \qquad (2.10,4a)$$

$$\xi_E = \arctan(-z_T/x_T) \qquad (2.10,4b)$$

3. BALSIM SOFTWARE DESCRIPTION — GENERAL

The dynamic model described in the previous chapter has been implemented in the BALSIM simulation package. All coding was carried out using IBM FORTRAN for the H-extended compiler. The package has been debugged and tested on the IBM 3033 computer, and has been used to predict the dynamic characteristics of the CRV7/BATS and ROBOT-9 vehicles.

The software is currently being installed on VAX11/780 and Honeywell DPS-8/70C computers.

The software consists of a MAIN program plus nine subroutines making up approximately 885 FORTRAN source statements. There are no subroutines or functions, other than these, that are not available in standard FORTRAN on-line libraries.

The software userbook is given in Appendix 1 with a source language listing of the package.

3.1 Software Capabilities

The dynamic model implemented with the BALSIM package has already been discussed in detail in the previous chapter. Its limitations will not be considered further here.

The package was developed with the objective of providing a convenient basis for inputting the characteristics of multistaged rocket vehicles and predicting their dynamic rigid body characteristics. By appropriately modifying the input data set, it provides for

1) nominal and off-nominal vehicle mass, inertia and thrust characteristics,
2) different motor types,
3) Mach number dependent aerodynamic characteristics,
4) structural production tolerances,
5) system failures (e.g. stage and fin failures),
6) moving launchers,
7) user specified initial conditions,
8) user specified payload characteristics,
9) tabular output in either metric or English units, and
10) multiple case runs.

As well, with minor software modification, response calculations may be stored on disk for subsequent use with other software (e.g. plotting software).

3.2 Software Limitations

In its current form the software is not intended for use in the following types of simulations:

1) Ballistic rocket vehicles with control surfaces.
2) Winged flight vehicles.
3) Nonrigid vehicles.
4) Simulations where the inertial, flat Earth approximations are invalid.
5) Vehicles where the staging process involves physically releasing rocket motor stages.

Limitations (1) and (5) may be removed with relatively minor alterations to the dynamic model of Chapter 2 with corresponding changes to the software.

3.3 Numerical Integration Algorithm

The numerical integration algorithm used to solve the system of ordinary differential equations describing the vehicle's dynamics is a fixed step-size, fourth order Runge-Kutta method (see Reference 7). Provision has been made for the specification of two step sizes, one for use during rocket motor burns, and the other for use during coasting flight. The latter technique was found to considerably reduce CPU time in certain simulations.

3.4 Software Testing and Execution Times

The BALSIM package has been used extensively to predict the performance and dynamic characteristics of the CRV7/BATS and ROBOT-9 vehicles. These predictions have been used to define the nominal, dispersion and safety-envelope characteristics (see References 8 and 9) of these vehicles.

Flight test data obtained early in the development of CRV7/BATS (see Reference 1) was used to update and validate the aerodynamic model that has been employed. More recent comparisons with flight test data have also proven to be satisfactory (also see Reference 1).

BALSIM predictions have also been evaluated for consistency by comparing results obtained using the equations of motion written in F_B with those written in F_B', with satisfactory results.

The execution CPU time of the package will depend on the computer used, on the step sizes chosen, and on the duration of the flight time simulated. For the IBM 3033 computer, the following CPU execution times were observed for simulations of the ROBOT-9 vehicle using a 0.05 second integration step size, and a 1.0 second tabulated output increment:

1) For 8 cases averaging 108 simulated flight seconds per case, the execution CPU time was 343.4 seconds, yielding a 0.4 seconds CPU execution time per simulated flight second ratio.
2) The compilation CPU time with the H-extended compiler was 25 seconds.
3) The linkage editor CPU time was 1.6 seconds.

This completes the general description of the BALSIM software package. Detailed user related data is given in Appendix 1.

4. SUMMARY

Six degree-of-freedom, rigid body equations of motion suitable for modeling the dynamic characteristics of multistaged, free-flight, ballistic rockets have been rigorously developed, and have been implemented in a FORTRAN software package called BALSIM. This package allows for modeling of vehicle thrust and structural asymmetries, time-varying mass and inertia characteristics, variable wind conditions, nonstandard atmospheric conditions, stage failures, and different rocket motor types.

The BALSIM package has been successfully used to predict the performance and dynamic characteristics of the CRV7/BATS and ROBOT-9 vehicles both with and without moving launchers. It will be adapted for use with the VAX11/780 and Honeywell DPS-8 computers in the near future.

REFERENCES

1. Coffey, C. and Markov, A., "A Rocket-Boosted Sea Launched Target System", **Third International Bristol RPV Conference**, 13 – 15 September, 1982.

2. Markov, A., **The Landing Approach in Variable Winds: Curved Glidepath Geometries and Worst-Case Wind Modeling**, UTIAS Report No. 254, University of Toronto, 1981.

3. Etkin, B., **Dynamics of Atmopsheric Flight**, John Wiley & Sons, Inc., Toronto, 1972.

4. McRuer, Duane, Ashkenas, Irving and Graham, Dunstan, **Aircraft Dynamics and Automatic Control**, Princeton University Press, New Jersey, 1973.

5. Anonymous, **Design of Aerodynamically Stablized Free Rockets Engineering Design Handbook**, AMC Pamphlet AMCP 706-280, U.S. Army Materiel Command, July 1968.

6. McCormick, Barnes W., **Aerodynamics, Aeronautics, and Flight Mechanics**, John Wiley & Sons, Toronto, 1979.

7. Gear, C.W., **Numerical Initial Value Problems in Ordinary Differential Equations**, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1971.

8. Schilling, K. and Markov, A., **CRV7/BATS Performance Scatter and Safety-Envelope (U)**, Suffield Memorandum No. 1070, Defence Research Establishment Suffield, December 1981, UNCLASSIFIED.

9. Markov, A., **ROBOT-9 Performance Scatter and Safety Envelope (U)**, Suffield Memorandum No. 1075, Defence Research Establishment Suffield, to be published 1983, UNCLASSIFIED.

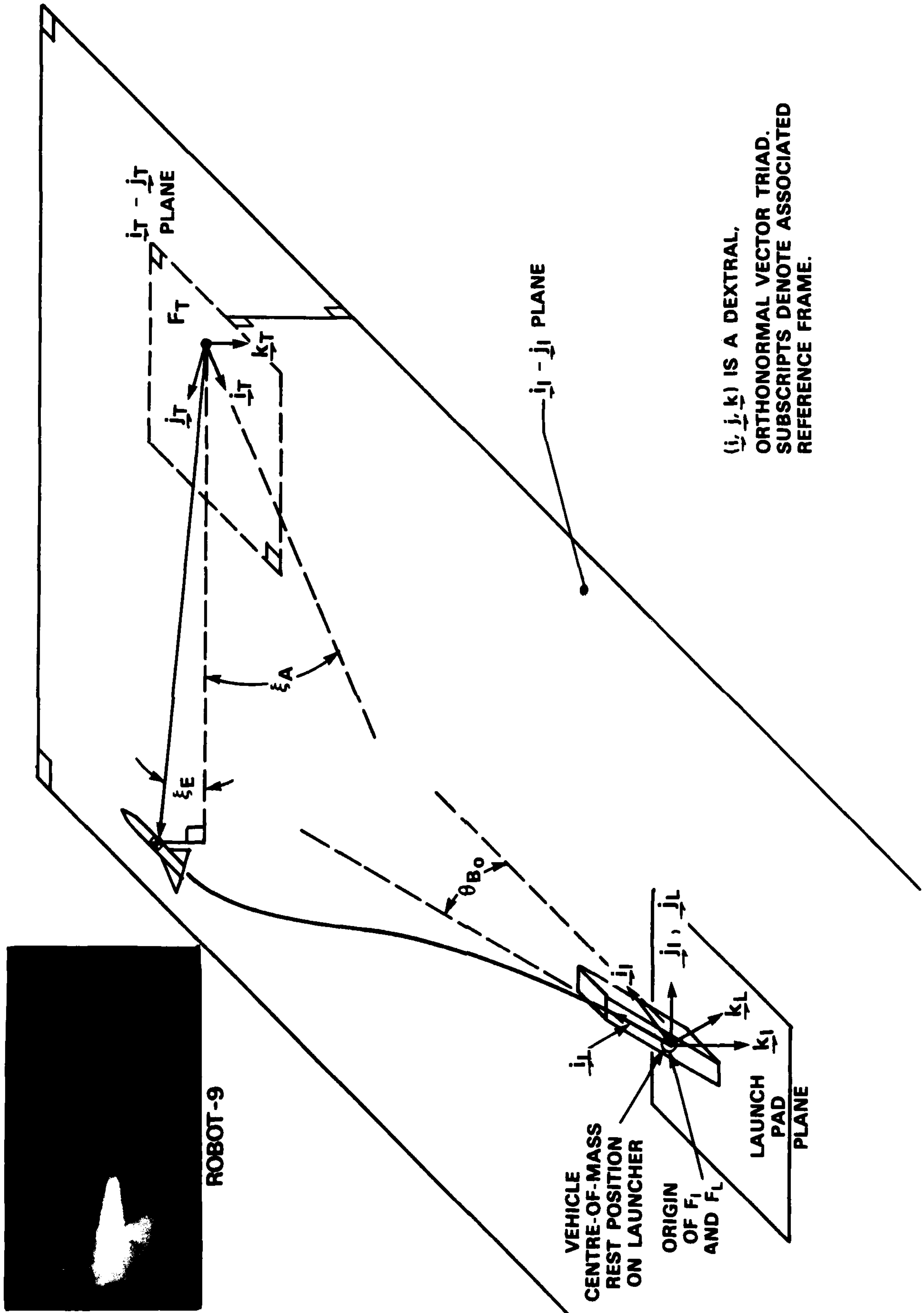

FIGURE 1

DEFINITION OF F_I, F_L, F_T

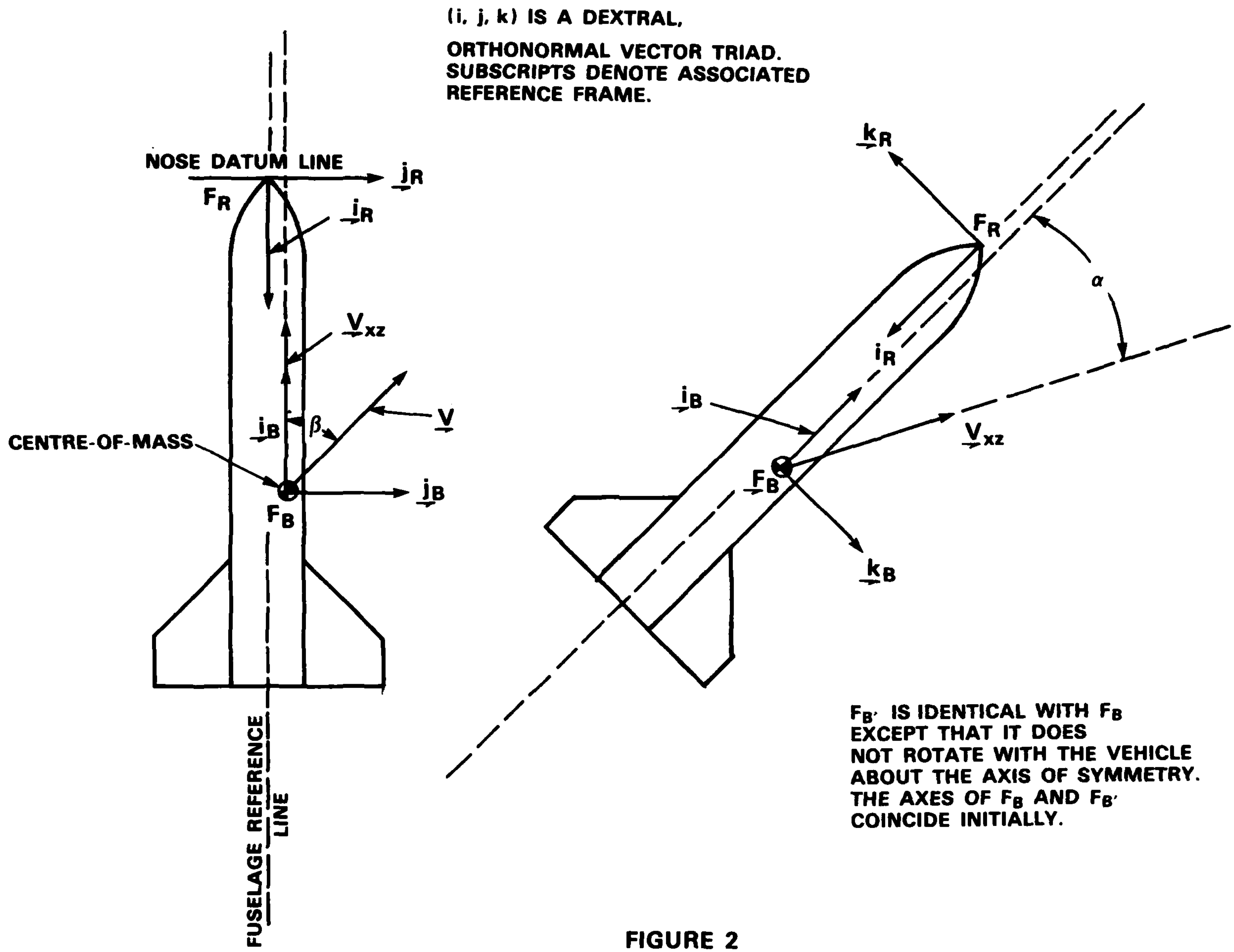

FIGURE 2

DEFINITION OF F_B, $F_{B'}$, F_R, α AND β

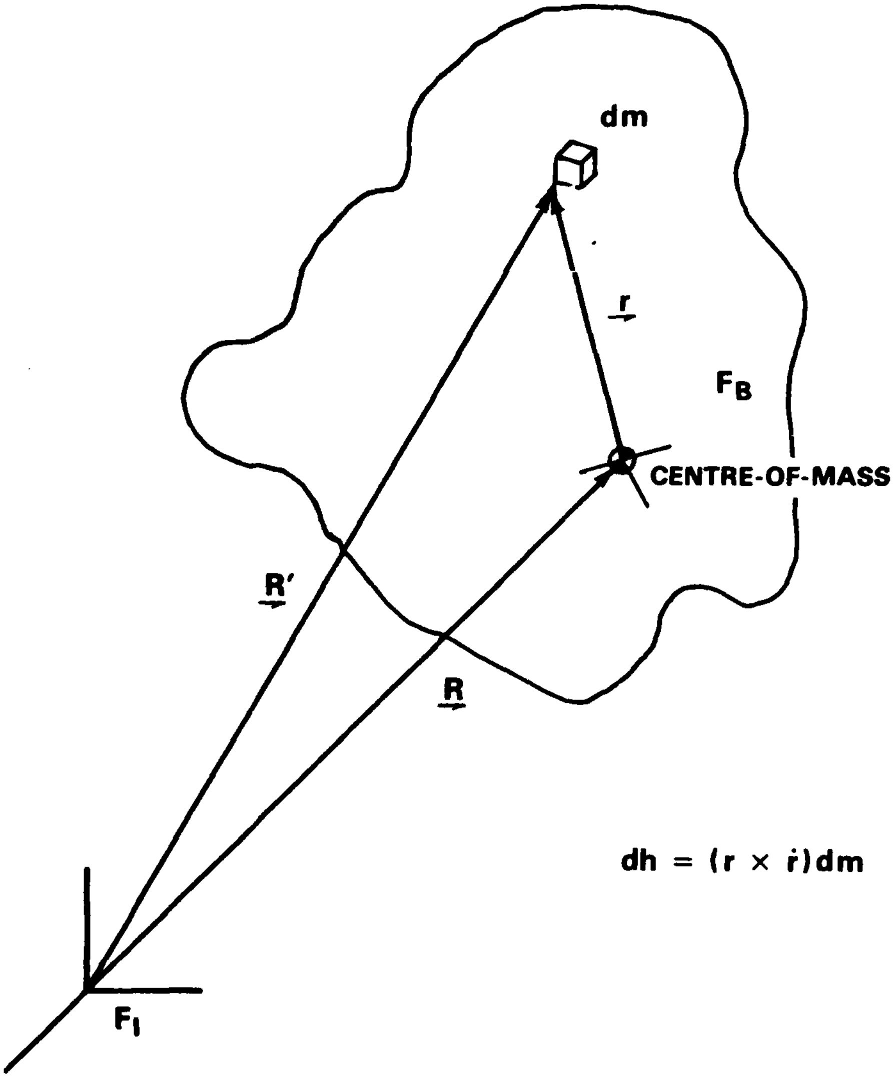

FIGURE 3

ANGULAR MOMENTUM CONTRIBUTION OF THE MASS ELEMENT dm

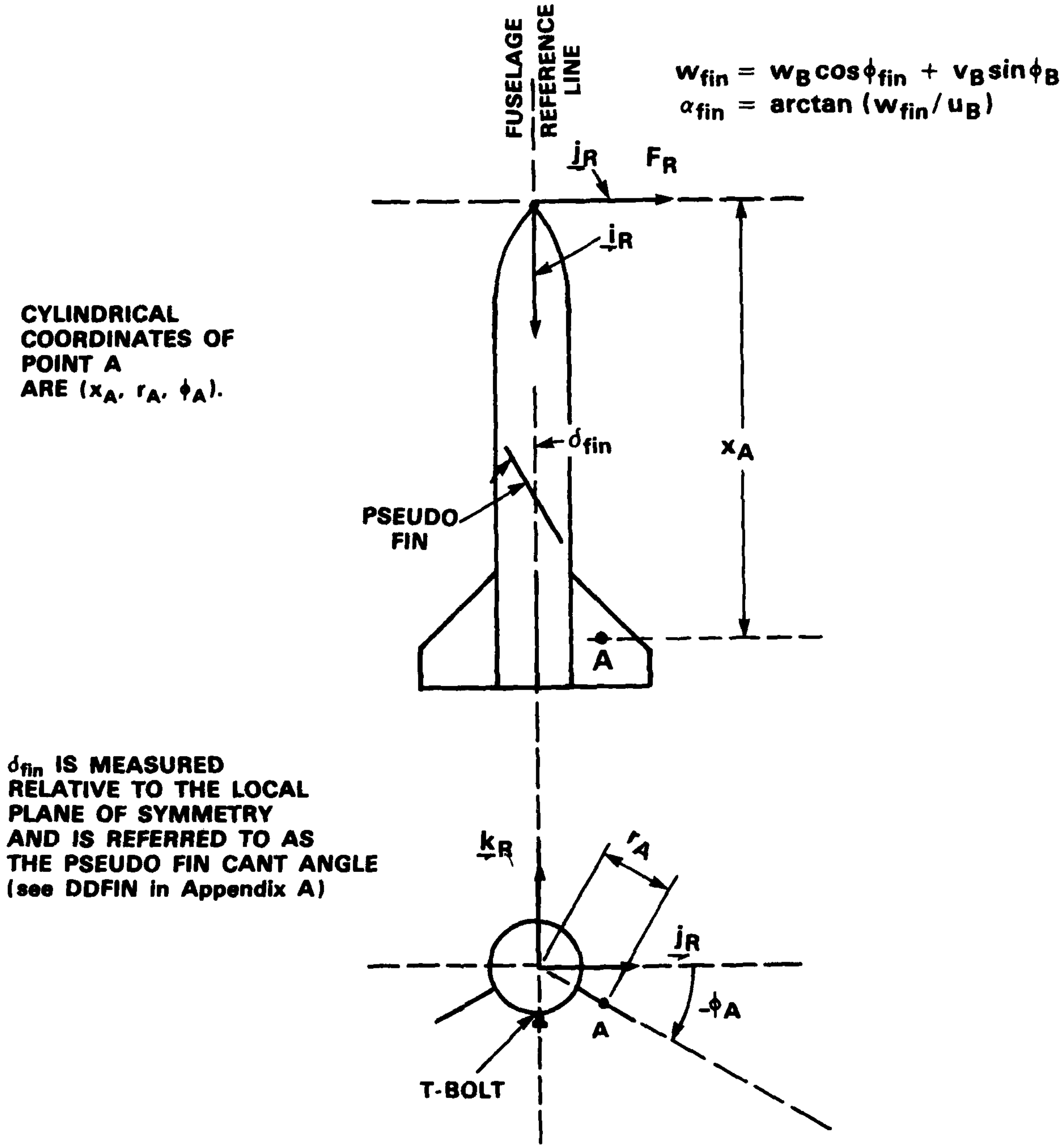

FIGURE 4

VEHICLE CYLINDRICAL COORDINATE SYSTEM

s = **distance vehicle has moved along launcher rail relative to rest position**

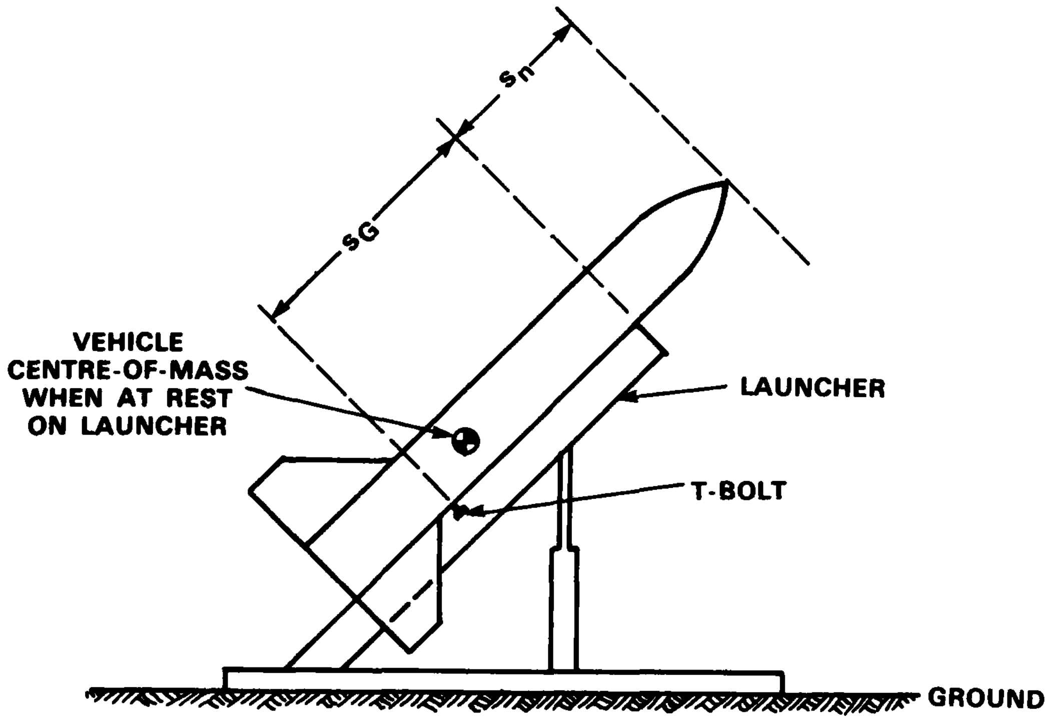

FIGURE 5

VEHICLE KINEMATIC CONSTRAINTS WHILE ON LAUNCHER GEOMETRY

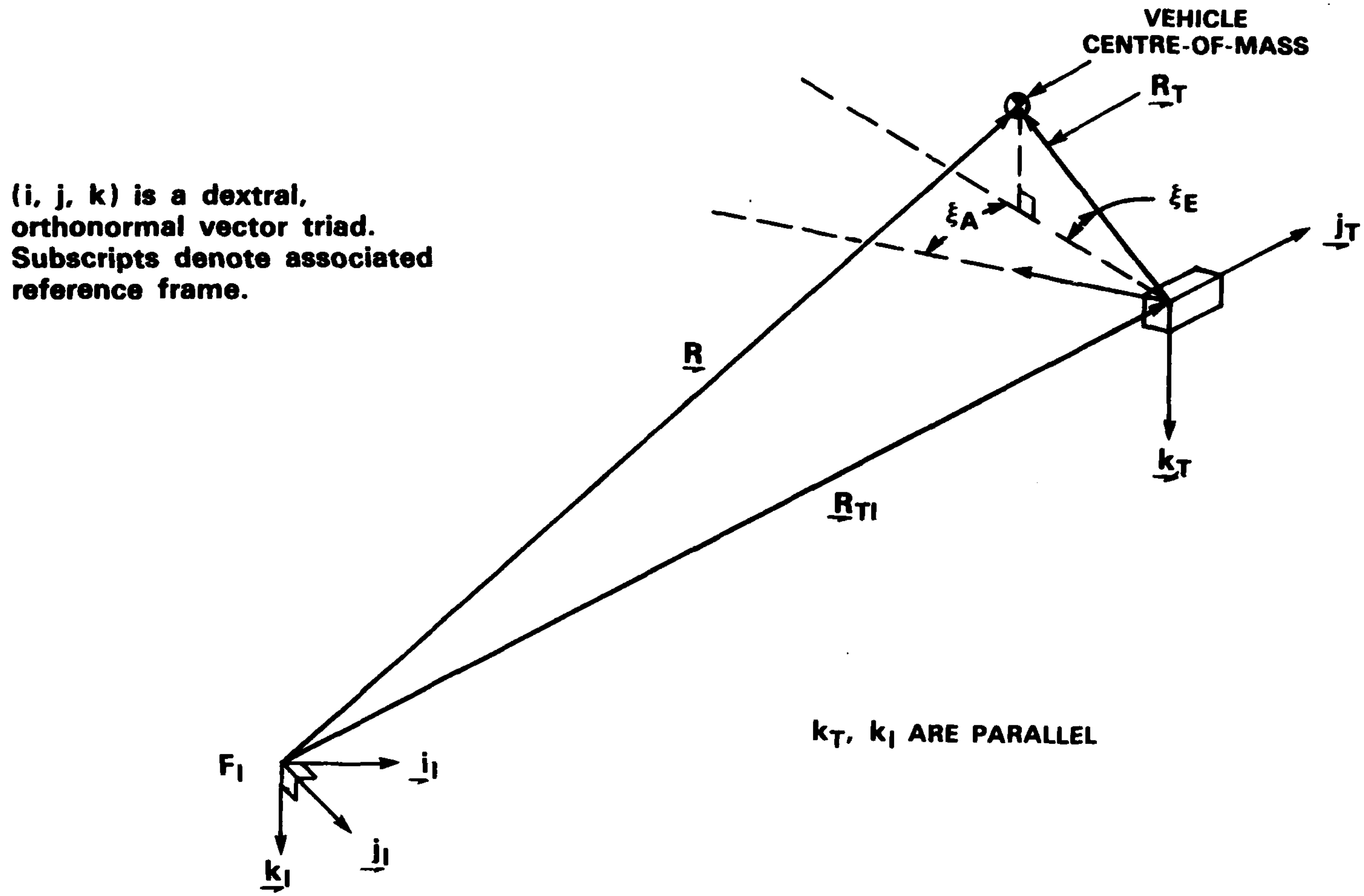

FIGURE 6

ASPECT ANGLE GEOMETRY

APPENDIX 1

BALSIM USERBOOK

APPENDIX 1

BALSIM USERBOOK

This appendix contains detailed user information for the BALSIM package. A source listing of the IBM H-Extended FORTRAN version of the package is given at the end of the Appendix.

As far as is possible, the user instructions are not system specific, and will in general apply to BALSIM's implementation on the VAX11/780 and the Honeywell DPS-8/70C computers as well as for the IBM 3033.

General background information on BALSIM is contained in Chapter 3 of the main text of the report. The equations of motion that are used are developed in detail in Chapter 2.

A1.1 PROGRAM UNITS

The BALSIM package consists of a MAIN program plus nine subroutines. These subroutines are as follows:

1) INPUT — input data from a card image file associated with unit number NIN.
2) INPUS — modifies input data as required for the NFIN cases.
3) RUNK1 — performs one step of the fourth order Runge-Kutta integration.
4) ATMOS — defines U.S. Standard Atmosphere (1962) and acceleration due to gravity characteristics as a function of altitude above sea level (ALT).
5) WIND — defines components of the wind velocity vector in body-axes as a function of altitude above sea level (ALT).
6) INTEQ — performs a linear interpolation of arrays TY1, TY2, TY3, whose common, possibly unevenly spaced abscissae are stored in the array TX.

7) INTPL — performs a linear interpolation of the array TY whose abscissae are stored in the array TX.

8) OUTPU — carries out tabular output as required by the parameter NCOUT.

9) OUTAB — carries out summary output for all cases run.

A1.2 INPUT DATA

The input data is stored on a card image file associated with unit number NIN.

The input data organization is summarized below, and may also be obtained from subroutine INPUT.

DATA ITEM	FORMAT
1. NOROL, NCOUT, NIOUT	7I2
2. DEL1, DEL2, TSTOP, ALTST, DELPT	8F10.2
3. SGUID, SFRLN	8F10.2
4. XIO, YIO, ALTO	8F10.2
5. UO, VO, WO	8F10.2
6. THETO, PSIO, PHIO	8F10.2
7. PO, QO, RO	8F10.2
8. CLP, CMQ, CLDA, DELTF, B, SREF	8F10.2
9. CLFIN, DDFIN, CPFIN, PHFIN, FIFAC	8F10.2
10. DXCP, FXCN, FXCD	8F10.2
11. NMACH	7I2
12. Repeat NMACH times:	
AMACH(I), CDOT(I), CNAT(I), XCPT(I)	8F10.2
13. WTAPE, XCGTP, RCGTP, PCGTP, XIXTP, XIYTP, DIYZ	8F10.2
14. WPLD, XCGPL, RCGPL, PCGPL	8F10.2
15. MTYPE	7I2

DATA ITEM (cont'd)	FORMAT
16. Repeat MTYPE times:	
WTE(I), WTPR(I), SPIMP(I), CTHR(I), CTHRT(I)	8F10.2
NTTHR(I)	7I2
TTHRT(1,I), TTHR(1,I)	8F10.2
•	•
•	•
•	•
TTHRT(NTTHR(I),I), TTHR(NTTHR(I),I)	8F10.2
17. NKICK	7I2
18. Repeat NKICK times:	
NTYPE(I), NMOT(I)	7I2
TIG(I), XTHR(I), RTHR(I), PTHR(I), DTHR(I), ATHR(I)	8F10.2
XEMPT(I), XPROP(I), REMPT(I), RPROP(I), PEMPT(I), PPROP(I)	8F10.2
TORQL(I), TCCD(I), TCCL(I), TCCP(I)	8F10.2
19. NALTW	7I2
20. If NALTW $\leqslant$ 0, omit. Otherwise repeat NALTW times:	
ALTW(I), WIXA(I), WIYA(I), WIZA(I)	8F10.2
21. NALTA	
22. If NALTA $\leqslant$ 0, omit. Otherwise repeat NALTA times:	
ALTA(I), TMPR(I), PRES(I)	8F10.2
23. NFIN	7I2

This input data is defined as follows:

1. NOROL = I

   a) I = 1 – F_B' equations of motion (axisymmetrical mass, inertia, aerodynamic and thrust characteristics).

   b) I = 0 – F_B equations of motion (axisymmetrical mass, inertia and aerodynamic characteristics).

   c) I = –1 – F_B equations of motion (general case).

2. NCOUT = IJ: Tabular output control parameter.
   a) I = 0, J $\geqslant$ 0 — Output tables 1 to J + 1 inclusive in imperial units.
   b) I = 1, J $\geqslant$ 0 — Output tables 1 to J + 1 inclusive in MKS units.
   c) I < 0 — Output no tables.
3. NIOUT = I: Input data output control parameter.
   a) I = 0 — No listing of input variables.
   b) I = 1 — Listing of input variables.
4. DEL1: Full Runge-Kutta integration step size with rocket motors off (seconds).
5. DEL2: Full Runge-Kutta integration step size with rocket motors on (seconds).
6. TSTOP: Maximum permissable simulation time (seconds).
7. ALTST: Altitude above launch point at which the simulation will end (ft).
8. DELPT: Time step for output tables (seconds), t_{max}/DELPT + 1 $\leqslant$ 200.
9. SGUID: s_G (see Figure 5, ft).
10. SFRLN: s_n (see Figure 5, ft).
11. XIO, YIO, ALTO: $x_I(0)$, $y_I(0)$, $h_{ASL}(0)$ (ft)
    a) $h_{ASL}(0)$ is the altitude above sea level of the centre-of-mass of the vehicle while at rest on the launcher.
12. UO, VO, WO: $u_{B_E}(0)$, $v_{B_E}(0)$, $w_{B_E}(0)$ (fps).
13. THETO, PSIO, PHIO: $\theta_B(0)$, $\psi_B(0)$, $\phi_B(0)$ (deg).

14. PO, VO, RO: $p_B(0)$, $v_B(0)$, $r_B(0)$ (deg/second).

15. CLP: C_{l_p} (relative to the diameter and cross-sectional area of the fuselage, 1/deg).

16. CMQ: C_{m_q} (relative to the diameter and cross-sectional area of the fuselage, 1/deg).

17. CLDA: $C_{l_{\delta fin}}$ (relative to the diameter and cross-sectional area of the fuselage, 1/deg).

18. DELTF: δ_{fin} (deg).

19. B: Fuselage diameter, b (ft).

20. SREF: Fuselage cross-sectional area, S (ft^2).

21. CLFIN: C_{L_α} pseudo fin, relative to S (1/deg).

22. DDFIN: $(\delta_{fin})_{pseudo}$ (deg).

23. CPFIN: $x_{ac_{fin}}$ (ft).

24. PHFIN: ϕ_{fin}, angular cylindrical coordinate of aerodynamic centre of the pseudo fin (see Figure 4, deg).

25. FIFAC: e_{fin}, pseudo fin efficiency factor, >0 positive fin, <0 negative fin.

26. DXCP: Vehicle aerodynamic centre correction term, XCPT = XCPT + DXCP (positive values move the aerodynamic centre further back on the vehicle, ft).

27. FXCN: Factor adjusting C_{N_α} data (normally FXCN = 1), i.e. $C'_{N_\alpha} = C_{N_\alpha} * FXCN$.

28. FXCD: Factor adjusting C_D data (normally FXCD = 1), i.e. $C'_D = C_D * FXCD$.

29. NMACH: The number of different Mach numbers at which aerodynamic data is given (NMACH $\leq$ 20).

30. AMACH: Mach number array of dimension 20.

31. CDOT: $C_D(M_a)$ array of dimension NMACH.

32. CNAT: $C_{N_\alpha}(M_a)$ array of dimension NMACH (1/deg).

33. XCPT: $x_{ac}(M_a)$ array of dimension NMACH (ft).

34. WTAPE: $m_{em}g$ (0), empty fuselage weight (lb).

35. XCGTP: x_{em} (x-coordinate of airframe (empty) centre-of-mass position, see Figure 4, ft).

36. RCGTP: r_{em} (radial cylindrical coordinate of airframe (empty) centre-of-mass positions, see Figure 4, ft).

37. PCGTP: ϕ_{em} (angular cylindrical coordinate of airframe (empty) centre-of-mass position, see Figure 4, deg).

38. XIXTP: $I^B_{xx_{em}}$ (slugs.ft²).

39. XIYTP: $I^B_{yy_{em}}$ (slugs.ft²).

40. DIYZ: Asymmetry factor ($I^B_{zz_{em}} = [1 + \text{DIYZ}]\, I^B_{yy_{em}}$).

41. WPLD: Weight payload (lbs).

42. XCGPL: x_{PL} (x-coordinate of payload centre-of-mass position, see Figure 4, ft).

43. RCGPL: r_{PL} (radial cylindrical coordinate of payload centre-of-mass position, ft).

44. PCGPL: ϕ_{PL} (angular cylindrical coordinate of payload centre-of-mass position, see Figure 4, deg).

45. MTYPE: Number of motor types (maximum of 5).

46. WTE(I): Motor weight empty (lb) of I-th motor type.

47. WTPR(I): Propellant weight (lb) of I-th motor type.

48. SPIMP(I): Specific impulse of I-th motor type, I_{sp_i} (sec).

49. NTHR(I): Number of thrust table points for the I-th motor type (maximum of 100).

50. CTHR(I): I-th motor type thrust correction factor (normally 1.0).

51. CTHRT(I): I-th motor type thrust table time correction factor (normally 1.0).

52. TTHRT (J,I): Time points for I-th motor type (do not have to be equally spaced, $1 \leq J \leq$ NTHR(I), secs).

53. THRT (J,I): Thrust levels for I-th motor type corresponding to time points in TTHRT (J,I) ($1 \leq J \leq$ NTHR(I), lbs).

54. NKICK: Number of stages (maximum of 10).

55. NTYPE(I): Motor type used in the I-th stage.

56. NMOT(I): Number of motors of type NTYPE(I) used in the I-th stage, NMOT(I) ≤ 3.

57. TIG(I): Time of ignition of the I-th stage (secs). The time of ignition of two successive stages may be identical, e.g. as would be the case if two different motor types are fired simultaneously.

58. XTHR(I): x-coordinate (in F_R) of the point at which the thrust acts for the 'key' motor of the I-th stage (see Figure 4, ft). If there is only one motor for the I-th stage, i.e. NMOT(I) = 1, then the coordinate is for that motor's thrust vector point of action. If NMOT(I) = 2, then the coordinate is for one of the motor's thrust vectors, and the program assumes that the second motor is axisymmetrical about the x-axis of F_R with respect to the first motor. If NMOT(I) = 3, then the coordinate is for one of the motors off the x-axis of F_R. The program then assumes that a second motor exists that is axisymmetrical about the x-axis of F_R with respect to the first motor, and that the third motor is on the axis of symmetry.

59. RTHR(I): Radial cylindrical coordinate for the I-th stage for the same 'key' motor as used for XTHR(I) (see Figure 4, ft).

60. PTHR(I): Angular cylindrical coordinate for the I-th stage for the same 'key' motor as used for XTHR(I) (see Figure 4, deg).

61. DTHR(I): Key rocket motor thrust vector reference frame Euler angle, see Appendix 2.

62. ATHR(I): Key rocket motor thrust vector reference frame Euler angle, see Appendix 2.

63. XEMPT(I): x-coordinate (in F_R) of the centre-of-mass of the empty motor case of the 'key' motor of the I-th stage (see Figure 4, ft).

64. REMPT(I): Radial cylindrical coordinate (in F_R) of the centre-of-mass of the empty motor case of the 'key' motor of the I-th stage (see Figure 4, ft).

65. PEMPT(I): Angular cylindrical coordinate (in F_R) of the centre-of-mass of the empty motor case of the 'key' motor of the I-th stage (see Figure 4, ft).

66. XPROP(I), RPROP(I), PPROP(I): Coordinates analogous to XEMP(I), REMPT(I), PEMPT(I) for the cylindrical coordinates (in F_R) for the centre-of-mass of the propellant of the 'key' motor of the I-th stage.

67. TORQL(I): Torque nondimensional coefficient about the thrust axis of 'key' motor of the I-th stage due to spiral grooves or vanes in the nozzel of the motor.

68. TCCD: Coefficient describing the rocket plume exhaust effect on the drag of the vehicle.

69. TCCL: Coefficient describing the rocket plume exhaust effect on the aerodynamic normal forces acting on the vehicle.

70. TCCP: Coefficient describing the rocket plume exhaust effect on the aerodynamic centre of the vehicle.

71. NALTW: Number of altitudes at which the wind velocity is given (NALTW ⩽ 100). If NALTW = 0, no wind data is expected.

72. ALTW(I): Altitudes above sea level at which wind velocity vector components are given (ft). These altitudes do not have to be evenly spaced.

73. WIXA(I): W_1 corresponding to ALTW(I) (fps).

74. WIYA(I): W_2 corresponding to ALTW(I) (fps).

75. WIZA(I): W_3 corresponding to ALTW(I) (fps).

76. NALTA: Number of altitudes at which atmospheric temperature and pressure data is given (NALTA ⩽ 100). If NALTA = 0, no temperature and pressure is expected.

77. ALTA(I): Altitudes above sea level at which atmospheric temperature and pressure data is given (ft). These altitudes do not have to be evenly spaced.

78. TEMPT(I): The % deviation from standard temperature corresponding to ALTA(I).

79. PRES(I) The % deviation from standard pressure corresponding to ALTA(I).

80. NFIN: Number of cases (NFIN ⩽ 100).

A1.3 SOFTWARE ALGORITHM

The algorithm used is as follows:

1) Input date through unit NIN using subroutine INPUT.

2) Set NSHOT = 1. Total number of cases for a given run is NFIN.

3) Using subroutine INPUS, redefine input dataset as appropriate for case NSHOT.

4) Output headers and results as appropriate for value of parameter NCOUT.

5) Compute starting parameters, including initial mass and inertia characteristics of the simulated vehicle.

6) Define values of ϱ, p_A, T_A and g appropriate to h_{ASL} with subroutine ATMOS.

7) Define aerodynamics.

8) Perform Runge-Kutta integration using subroutine RUNK1. Use step size DEL1 if rocket motors are off, and DEL2 if rocket motors are on.

9) Output Table 1 data if NCOUT $\geqslant 0$ and if an output time point is reached (at DELPT time intervals).

10) Store results at output time points for all relevant parameters in the array 0(I, J), where I is the parameter and J is the output time step.

11) Compare outputted results with previous outputted results to get current maximum Mach number, altitude and dynamic pressure data.

12) Go to Step 6 and repeat until either $-z_I - \text{ALTST} - 0.5\dot{z}_I\text{DELT} \leqslant 0$ or t $\geqslant$ TSTOP. Here DELT = DEL1 or DEL2 (see Step 8).

13) Output tabular data based on data stored in array 0 as required by parameter NCOUT. Convert to and output in metric units if NCOUT $\geqslant 11$, otherwise use Imperial units.

14) If case is finished, as determined by Step 12, store case summary data (range, apogee, flight-time, apogee-time, maximum Mach number, and maximum dynamic pressure). Output summary data for case just completed.

15) NSHOT = NSHOT + 1. If NSHOT $\leqslant$ NFIN, go to Step 3. Otherwise, output case summary data (see Step 14) and STOP.

A1.4 GENERAL NOTES

A1.4.1 Integration Algorithm Numerical Instability

The Runge-Kutta algorithm will be unstable if DEL1 and DEL2 are chosen too large, a situation that will usually result in an overflow condition. This problem may be resolved by making the integration step sizes DEL1 and DEL2 smaller.

A1.4.2 Simultaneous Firing of Two Different Stages

The time of ignition of two successive stages may be identical. This would be required if two different motor types are fired simultaneously, or if more than three motors of one type are used in a given stage (see the restrictions placed on NMOT(I) in Section A1.2). Also, thrust asymmetries must be handled by specifying the characteristics of each motor in a given stage individually, i.e., by treating them as separate stage firings with the same ignition time.

A1.4.3 TCCD, TCCL, TCCP

These coefficients are given as inputs in order to take into account rocket plume exhaust effects on the aerodynamics of the vehicle (see Section A1.2). For CRV7/BATS and ROBOT-9 type vehicles, these exhaust effects on lift, side force and aerodynamic centre characteristics are negligible, and thus TCCL = TCCP = 1.0.

Rocket motors fired in the base of the vehicle will significantly reduce base drag. For CRV7/BATS and ROBOT-9 a representative value for TCCD for such a stage firing is TCCD = 0.8.

A1.4.4 Vehicle Symmetry Assumptions

Whenever it is the intent to use the F_B' equations (NOROL = 1, see Section A1.2 and also Section 2.2), all vehicle mass, inertia and thrust characteristics must be specified with axisymmetry about the x-axis of F_B. Otherwise, NOROL = 0 or −1 must be used in order to invoke the use of the F_B equations of motion.

For NOROL = 0, axisymmetric mass, inertia and aerodynamic characteristics must be inputted. General thrust characteristics may be used.

For NOROL = −1, no symmetry constraints are placed on inputted mass, inertia, aerodynamic and thrust characteristics.

A1.4.5 Output Tables

The output tables are generated as determined by the value of the parameter NCOUT (see Section A1.2). The contents of each table may be determined from the output headers in the subroutine OUTPU (see Section A1.5 to follow).

Table 1 contains the aspect angle presentations of the target relative to an observer located at a prespecified point relative to the launch site reference frame F_I. The program assumes that the observer is located at (x_{T_I}, 0, 0) (see Section 2.10). x_{T_I} may be varied by changing the appropriate statements in the MAIN program, i.e., the statements defining array elements 0(59, IOUT) and 0(60, IOUT).

A1.5 BALSIM SOURCE LISTING

The following listing is the IBM FORTRAN IV (H-Extended Compiler) version of the BALSIM package. It is recommended that the AUTODBL feature is used so that all real arithmetic is carried out in double precision.

```
C
C     PROGRAM  B A L S I M, VERSION 2 (IBM SYSTEM 370)   OCTOBER 1982
C
      COMMON /INDAT/ NOROL, DEL1, TSTOP, ALTST, DELPT, NCOUT, NIOUT, DEL2,
     *SLAUN, SGUID, SFRLN, XIO, YIO, ALTO, UO, VO, WO, THETO, PSIO, PHIO, PO QO RO
     *CLP, CMQ, CLDA, DELTF, B, NMACH, AMACH(20), CDOT(20), CNAT(20), XCPT(20
     *WTAPE, XCGTP, RCGTP, PCGTP, XIXTP, XIYTP, DIYZ, WPLD, XCGPL, RCGPL, PCGPL,
     *CLFIN, CPFIN, DDFIN, PHFIN, FIFAC, DXCP, FXCN, FXCD,
     *SREF,
     *MTYPE, WTE(5), WTPR(5), SPIMP(5), CTHR(5), CTHRT(5), NTTHR(5),
     *TTHRT(100,5), TTHR(100,5), COR1(5), COR2(5),
     *NKICK, NTYPE(10), NMOT(10), XTHR(10), ATHR(10), DTHR(10), PTHR(10)
     *XEMPT(10), REMPT(10), PEMPT(10), XPROP(10), RPROP(10), PPROP(10),
     *TIG(10), RTHR(10), TORQL(10), TCCD(10), TCCL(10), TCCP(10),
     *NALTW, ALTW(100), WIXA(100), WIYA(100), WIZA(100),
     *NALTA, ALTA(100), TMPR(100), PRES(100),
     * NFIN
      DIMENSION INIG(10), CFL(10), CFM(10), CFN(10), CFX(10), CFY(10), CFZ(10)
     *, WTS(10), WTSD(10), O(60,200), A(3,3), YPROP(10), ZPROP(10)
      DIMENSION QBMAX(100), XMMAX(100), APOGE(100), TAPOG(100), RANGE(100),
     *FLTIM(100), OO(7)
      DATA CNVRG, CNVGR, GRVO, NIN, NOUT
     *    /57.296, 0.017453, 32.174, 5, 6/
C
C      INITIAL DATA INPUT
C
      NSHOT = 0
      CALL INPUT(NIN, NOUT)
C
      DO 999 NSHOT=1, NFIN
C
C     VARIABLE INPUT FOR EACH SHOT
C
      CALL INPUS(NIN, NOUT, NSHOT, O, IOUT)
C
C     OPTIONAL ON-LINE OUTPUT (NCOUT.GE.0)
C
      IF(NCOUT) 510, 511, 511
  511 WRITE(NOUT, 100) NSHOT
      IF(NCOUT-10) 520, 521, 521
  521 WRITE(NOUT, 101)
      GOTO 510
  520 WRITE(NOUT, 102)
  510 CONTINUE
C
C       INITIAL CONDITIONS
C
      U = UO + 0.0001
      V = VO
      W = WO
      P = PO * CNVGR
      Q = QO * CNVGR
      R = RO * CNVGR
      PHI = PHIO    * CNVGR
      THET = THETO * CNVGR
      PSI  = PSIO  * CNVGR
      XI = XIO
      YI = YIO
      ZI = 0.
C
      WGTO = WTAPE + WPLD
```

```
      XCGTO = WTAPE*XCGTP + WPLD*XCGPL
      YCGTP = RCGTP*COS(PCGTP*CNVGR)
      ZCGTP = RCGTP*SIN(PCGTP*CNVGR)
      YCGPL = RCGPL*COS(PCGPL*CNVGR)
      ZCGPL = RCGPL*SIN(PCGPL*CNVGR)
      YCGTO = WTAPE*YCGTP + WPLD*YCGPL
      ZCGTO = WTAPE*ZCGTP + WPLD*ZCGPL
      XIXO = XIXTP*GRVO + WTAPE*RCGTP**2 + WPLD*RCGPL**2
      XIYO = XIYTP*GRVO + WTAPE*(XCGTP**2 + ZCGTP**2)
     *      + WPLD*(XCGPL**2 + ZCGPL**2)
      XIZO = XIYTP*(1.+DIYZ)*GRVO + WTAPE*(XCGTP**2 + YCGTP**2)
     *      + WPLD*(XCGPL**2 + YCGPL**2)
      XIXYO =-WTAPE*XCGTP*YCGTP - WPLD*XCGPL*YCGPL
      XIXZO = WTAPE*XCGTP*ZCGTP + WPLD*XCGPL*ZCGPL
      XIYZO =-WTAPE*YCGTP*ZCGTP - WPLD*YCGPL*ZCGPL
C
      DO 500 I=1,NKICK
      INIG(I) = 1
      DTH=DTHR(I)*CNVGR
      SDTH=DSIN(DTH)
      CDTH=DCOS(DTH)
      ATH  =ATHR(I)*CNVGR
      SATH = SIN(ATH)
      CATH = COS(ATH)
      PTH = PTHR(I)*CNVGR
      SPTH = SIN(PTH)
      CPTH = COS(PTH)
      CFX(I) =  1.-FLOAT(NMOT(I)/3)/FLOAT(NMOT(I))
      CFX(I) = CDTH*CATH*CFX(I)+1.-CFX(I)
      CFY(I) =-CDTH*SATH*SPTH-SDTH*CPTH
      CFZ(I) =-CDTH*SATH*CPTH+SDTH*SPTH
      CFL(I) = TORQL(I)*CFX(I)
      CFM(I) = CFZ(I)*XTHR(I) - RTHR(I)*SPTH*CFX(I) + TORQL(I)*CFY(I)
      CFN(I) =-CFY(I)*XTHR(I) - RTHR(I)*CPTH*CFX(I) + TORQL(I)*CFZ(I)
C
      K = NTYPE(I)
      FNMOT = FLOAT(NMOT(I))
      FSMOT = FLOAT(1+NMOT(I)/2)/FNMOT
      YPROP(I) = RPROP(I)*COS(PPROP(I)*CNVGR)
      ZPROP(I) = RPROP(I)*SIN(PPROP(I)*CNVGR)
      YEMPT = REMPT(I)*COS(PEMPT(I)*CNVGR)
      ZEMPT = REMPT(I)*SIN(PEMPT(I)*CNVGR)
      WTS(I) = WTPR(K)*FNMOT
      WTSD(I)=0.0
      WGTO = WGTO + WTE(K)*FNMOT
      XCGTO = XCGTO + WTE(K)*XEMPT(I)*FNMOT
      XIXO = XIXO + WTE(K)*REMPT(I)**2*FNMOT*FSMOT
      XIYO = XIYO + WTE(K)*(XEMPT(I)**2 + ZEMPT**2*FSMOT)*FNMOT
      XIZO = XIZO + WTE(K)*(XEMPT(I)**2 + YEMPT**2*FSMOT)*FNMOT
      IF(NMOT(I)-1) 519,519,500
  519 CONTINUE
      YCGTO = YCGTO + WTE(K)*YEMPT*FNMOT
      ZCGTO = ZCGTO + WTE(K)*ZEMPT*FNMOT
      XIXYO = XIXYO - WTE(K)*XEMPT(I)*YEMPT*FNMOT
      XIXZO = XIXZO + WTE(K)*XEMPT(I)*ZEMPT*FNMOT
      XIYZO = XIYZO - WTE(K)*YEMPT*ZEMPT*FNMOT
  500 CONTINUE
C
C     INITIAL AERODYNAMICS
C
      PHIFR = PHFIN*CNVGR
```

```
      SIPHF = SIN(PHIFR)
      COPHF = COS(PHIFR)
      CYFIO = -CLFIN*DDFIN*SIPHF
      CZFIO = -CLFIN*DDFIN*COPHF
      CYFIA = -FIFAC*CLFIN*SIPHF*CNVRG
      CZFIA = -FIFAC*CLFIN*COPHF*CNVRG
C
      RHFCT = 1
      VSFCT = 1.
      IF(NALTA-1) 501, 502, 504
  501 CONTINUE
      GOTO 504
  502 N1 = 0
      CALL ATMOS(RHO, PP, TK, VS, GRAV, ALTO, TMPR, PRES, ALTA, N1, RHFCT, VSFCT)
      RHFCT = PRES(1)*TK/(PP*TMPR(1))
      VSFCT = SQRT(TMPR(1)/TK)
  504 CONTINUE
      GBMAX(NSHOT) = 0.
      XMMAX(NSHOT) = 0.
      APOGE(NSHOT) = 0
      TIME = 0.
      PTIME = 0.
      XIALP = 0
      IOUT = 0
      KILL = 0
      DELT = DEL2
      KUTTA = 4
      GOTO 4
    1 KUTTA = KUTTA + 1
      GOTO (3, 2, 3, 2) KUTTA
    2 CONTINUE
      TIME = TIME + DELT * 0.5
    3 CALL RUNK1(KUTTA, DELT, U, UD, 1)
      CALL RUNK1(KUTTA, DELT, V, VD, 2)
      CALL RUNK1(KUTTA, DELT, W, WD, 3)
      CALL RUNK1(KUTTA, DELT, P, PD, 4)
      CALL RUNK1(KUTTA, DELT, Q, QD, 5)
      CALL RUNK1(KUTTA, DELT, R, RD, 6)
      CALL RUNK1(KUTTA, DELT, PHI, PHID, 7)
      CALL RUNK1(KUTTA, DELT, THET, THETD, 8)
      CALL RUNK1(KUTTA, DELT, PSI, PSID, 9)
      CALL RUNK1(KUTTA, DELT, XI, XDI, 10)
      CALL RUNK1(KUTTA, DELT, YI, YDI, 11)
      CALL RUNK1(KUTTA, DELT, ZI, ZDI, 12)
      DO 15 I = 1, NKICK
      J = I+12
   15 CALL RUNK1 (KUTTA, DELT, WTS(I), WTSD(I), J)
      CALL RUNK1 (KUTTA, DELT, XIALP, XIALPD, 24)
    4 CONTINUE
      PHI = AMOD(PHI, 6.28318531)
      STH=SIN(THET)
      XIALPD = GRVO*STH
      CTH=COS(THET)
      SPS=SIN(PSI)
      CPS=COS(PSI)
      SPH=SIN(PHI)
      CPH=COS(PHI)
      A(1,1)=CTH*CPS
      A(1,2)=CTH*SPS
      A(1,3)=-STH
      A(2,1)=SPH*STH*CPS-SPS*CPH
```

```
      A(2,2)=SPS*STH*SPH+CPS*CPH
      A(2,3)=SPH*CTH
      A(3,1)=CPS*CPH*STH+SPS*SPH
      A(3,2)=SPS*CPH*STH-CPS*SPH
      A(3,3)=CPH*CTH
      XDI = A(1,1)*U+A(2,1)*V+A(3,1)*W
      YDI = A(1,2)*U+A(2,2)*V+A(3,2)*W
      ZDI = A(1,3)*U+A(2,3)*V+A(3,3)*W
      ALT=-ZI + ALTO
      CALL ATMOS(RHO,POMP,TKELV,VS,GRAV,ALT,TMPR,PRES,ALTA,NALTA,
     *RHFCT,VSFCT)
      URL = U
      VRL = V
      WRL = W
      IF(NALTW)14,18,14
   14 CALL WIND(WIXA,WIYA,WIZA,ALTW,NALTW,ALT,A,WX,WY,WZ)
      URL = U-WX
      VRL = V-WY
      WRL = W-WZ
18    VR = SQRT(URL*URL+VRL*VRL+WRL*WRL)
      XMACH=VR/VS
      QBAR=0.5*RHO*VR*VR
      CDRAG = 1.0
      IF(IPASS) 913,913,912
  913 ALPHA= 0.0
      BETA = 0.0
      ALPHD = 0.0
      BETAD = 0.0
      IPASS = 1
      GO TO 914
  912 ALPHA = ATAN (WRL/URL)
      BETA = ATAN (VRL/SQRT(URL*URL+WRL*WRL))
      ALPHD=(WD/VR-UD*WD/(VR*VR))*COS(ALPHA)*COS(ALPHA)
      TEMP = SQRT (URL*URL+WRL*WRL)
      BETAD=(TEMP*VD-VRL*(URL*UD+WRL*WD)/TEMP)/(TEMP*TEMP*COS(BETA))
      WFIN = WRL*COPHF + VRL*SIPHF
      ALFIN = ATAN(WFIN/URL)
  914 CONTINUE
C
C     THRUST,MASS,C.G,MOM.IN ARE COMPUTED
C
      GOTO (390,300,390,300),KUTTA
  300 FEX = 0.
      FEY = 0
      FEZ = 0.
      EML = 0.
      EMM = 0.
      EMN = 0.
      LOMO = 0
      THRT = 0.
      XIDX = 0.
      XIDY = 0.
      XIDZ = 0.
      SWD = 0.
      SWDX = 0.
      SWDY = 0.
      SWDZ = 0.
      SWDX2 = 0.
      SWDY2 = 0.
      SWDZ2 = 0.
      LONSY = 0
```

```
C
      IF(NOROL) 310,311,311
  310 CONTINUE
      XIDXY = 0.
      XIDXZ = 0.
      XIDYZ = 0.
      SWDXY = 0.
      SWDXZ = 0.
      SWDYZ = 0.
  311 CONTINUE
C
      WGHT = WGTO
      XCGT = XCGTO
      XIX  = XIXO
      XIY = XIYO
      XIZ  = XIZO
C
      YCGT = YCGTO
      ZCGT = ZCGTO
      XIXY = XIXYO
      XIXZ = XIXZO
      XIYZ = XIYZO
C
C     SUM OVER ALL KICKS
C
      DO 308 I=1,NKICK
      WTSD(I)=0.0
      IF(TIME-TIG(I)) 301,302,302
  302 K=NTYPE(I)
      L=NTTHR(K)
      IF(TIME-TIG(I)-TTHRT(L,K)*CTHRT(K))304,304,301
  304 T =(TIME-TIG(I))/CTHRT(K)
      LOMO = 1
      LONSY = 2
      FSMOT = FLOAT(1+NMOT(I)/2)/FLOAT(NMOT(I))
      CALL INTPL (TTHR(1,K),TTHRT(1,K),THR,T,INIG(I),L)
      THR = THR * FLOAT(NMOT(I))*CTHR(K)
      THRT = THRT +THR
      WTSD(I) = -THR/(CTHRT(K)*SPIMP(K)*CTHR(K))
      XMASD = -WTSD(I)/GRVO
      SMASD = XMASD*FSMOT
      CDRAG = TCCD(I)
C
      FEX = FEX + THR*CFX(I)
      EML = EML + THR*CFL(I)
C
      SWD = SWD + WTSD(I)
      DWDX = WTSD(I)*XPROP(I)
      DWDY = WTSD(I)*YPROP(I)
      DWDZ = WTSD(I)*ZPROP(I)
      SWDX = SWDX + DWDX
      SWDY = SWDY + DWDY*FSMOT
      SWDZ = SWDZ + DWDZ*FSMOT
      SWDX2 = SWDX2 + DWDX*XPROP(I)
      SWDY2 = SWDY2 + DWDY*YPROP(I)*FSMOT
      SWDZ2 = SWDZ2 + DWDZ*ZPROP(I)*FSMOT
      IF(NMOT(I)-1) 301,307,301
  307 FEY = FEY +THR*CFY(I)
      FEZ = FEZ +THR*CFZ(I)
      EMM = EMM +THR*CFM(I)
      EMN = EMN +THR*CFN(I)
```

```
      SWDXY = SWDXY + DWDX*YPROP(I)
      SWDXZ = SWDXZ + DWDX*ZPROP(I)
      SWDYZ = SWDYZ + DWDY*ZPROP(I)
      LONSY = 1
  301 CONTINUE
C
      WGHT = WGHT + WTS(I)
      XCGT = XCGT + WTS(I)*XPROP(I)
      XIX  = XIX + WTS(I)*RPROP(I)**2*FSMOT
      XIY  = XIY + WTS(I)*(XPROP(I)**2 + ZPROP(I)**2*FSMOT)
      XIZ = XIZ + WTS(I)*(XPROP(I)**2 + YPROP(I)**2*FSMOT)
C
      IF(NMOT(I)-1) 320,320,308
  320 XIXY = XIXY - WTS(I)* XPROP(I)*YPROP(I)
      XIXZ = XIXZ + WTS(I)* XPROP(I)*ZPROP(I)
      XIYZ = XIYZ - WTS(I)* YPROP(I)*ZPROP(I)
      YCGT = YCGT + WTS(I)*YPROP(I)
      ZCGT = ZCGT + WTS(I)*ZPROP(I)
  308 CONTINUE
C
      IF(NOROL) 331,330,330
C
C     SYMMETRICAL BODY
C
  330 XMASS = WGHT/GRVO
      YCGT = O.
      ZCGT = O.
      XCGT = XCGT/WGHT
      XIX =  XIX/GRVO
      XIY =  XIY/GRVO - XMASS*XCGT**2
      XIZ =  XIZ/GRVO - XMASS*XCGT**2
      XIDX = SWDY2 + SWDZ2
      XIDY = SWDX2 + SWDZ2 + SWD*XCGT**2 - 2.*SWDX*XCGT
      XIDZ = SWDX2 + SWDY2 + SWD*XCGT**2 - 2.*SWDX*XCGT
      XIDX = XIDX/GRVO
      XIDY = XIDY/GRVO
      XIDZ = XIDZ/GRVO
      EMM = EMM - FEZ*XCGT
      EMN = EMN + FEY*XCGT
      GOTO 390
C
C
C     UNSYMMETRICAL BODY
C
  331 XMASS = WGHT/GRVO
      XCGT = XCGT/WGHT
      YCGT = YCGT/WGHT
      ZCGT = ZCGT/WGHT
      CGZ  = XCGT**2 + YCGT**2
      CGY   = XCGT**2 + ZCGT**2
      CGX   = YCGT**2 + ZCGT**2
      CGXY  =- XCGT*YCGT
      CGXZ  = XCGT*ZCGT
      CGYZ  = -YCGT*ZCGT
C
      XIX = XIX/GRVO -XMASS*CGX
      XIY = XIY/GRVO -XMASS*CGY
      XIZ = XIZ/GRVO -XMASS*CGZ
      XIXY = + XIXY/GRVO - XMASS*CGXY
      XIXZ = + XIXZ/GRVO - XMASS*CGXZ
      XIYZ = + XIYZ/GRVO - XMASS*CGYZ
```

```
C
      SWCX = SWDX*XCGT
      SWCY = SWDY*YCGT
      SWCZ = SWDZ*ZCGT
C
      XIDX = SWDY2 + SWDZ2 + SWD*CGX - 2.*(SWCY+SWCZ)
      XIDY = SWDZ2 + SWDY2 + SWD*CGY - 2.*(SWCX+SWCZ)
      XIDZ = SWDY2 + SWDZ2 + SWD*CGZ - 2.*(SWCX+SWCY)
      XIDXY =-SWDXY + SWD*CGXY + XCGT*SWDY + YCGT*SWDX
      XIDXZ = SWDXZ + SWD*CGXZ - XCGT*SWDZ - ZCGT*SWDX
      XIDYZ =-SWDYZ + SWD*CGYZ + YCGT*SWDZ + ZCGT*SWDY
      XIDX = XIDX/GRVO
      XIDY = XIDY/GRVO
      XIDZ = XIDZ/GRVO
      XIDXY = XIDXY/GRVO
      XIDXZ = XIDXZ/GRVO
      XIDYZ = XIDYZ/GRVO
C
C     CALCULATION OF INVERSE TENSOR OF INERTIA
C
      YIX = XIY*XIZ - XIYZ**2
      YIXY = XIXZ*XIYZ + XIXY*XIZ
      YIXZ = XIXY*XIYZ + XIXZ*XIY
      YNEN = XIX*YIX + XIXY*YIXY - XIXZ*YIXZ
      YIX = YIX/YNEN
      YIXY = YIXY/YNEN
      YIXZ = YIXZ/YNEN
      YIY = (XIX*XIZ - XIXZ**2)/YNEN
      YIYZ = (XIXZ*XIXY + XIYZ*XIX)/YNEN
      YIZ = (XIX*XIY - XIXY**2)/YNEN
C
C THRUST MOMENTS
C
      EML = EML - FEZ*YCGT - FEY*ZCGT
      EMM = EMM - FEZ*XCGT + FEX*ZCGT
      EMN = EMN + FEY*XCGT + FEX*YCGT
C
  390 CONTINUE
C
C     AERODYNAMICS
C
      QSREF = QBAR*SREF
      QSVRF = QSREF*B*B*CNVRG/(2.*VR)
      CALL INTEQ (CDOT,CNAT,XCPT,AMACH,CDO,CNA,XCP,XMACH,NMACH)
C
      XCP =XCP + DXCP
      FAX = -CDO *QSREF*CDRAG*FXCD
      FAY = -CNA *BETA*QSREF*CNVRG*FXCN
      FAZ = -CNA*ALPHA*QSREF*CNVRG*FXCN
      DDFAY = QSREF*(CYFIA*ALFIN+CYFIO)
      DDFAZ = QSREF*(CZFIA*ALFIN+CZFIO)
C
      AL = CLP*P*QSVRF + CLDA*DELTF*QSREF*B - FAY*ZCGT - FAZ*YCGT
      AM = FAZ*(XCP-XCGT) + CMQ*Q*QSVRF +FAX*ZCGT + DDFAZ*(CPFIN-XCGT)
      AN =-FAY*(XCP-XCGT) + CMQ*R*QSVRF +FAX*YCGT - DDFAY*(CPFIN-XCGT)
C
      FAY = FAY + DDFAY
      FAZ = FAZ + DDFAZ
C
      FXT=FAX+FEX
      FYT=FAY+FEY
```

```
      FZT=FAZ+FEZ
      TL = AL + EML
      TM = AM + EMM
      TN = AN + EMN
C
C
C     DYNAMICS
C
      THETD = Q*CPH-R*SPH
      PSID=(Q*SPH+R*CPH)/CTH
      IF(NOROL) 403,401,402
C
  401 PHID = P + PSID*STH
      UD=FXT/XMASS-W*Q+V*R+GRAV*A(1,3)
      VD=FYT/XMASS+W*P-U*R+GRAV*A(2,3)
      WD=FZT/XMASS+U*Q-V*P+GRAV*A(3,3)
C
      HCA = (XIZ-XIY)*Q*R
      HCB = (XIX-XIZ)*P*R
      HCC = (XIY-XIX)*P*Q
C
      HDA = XIDX*P
      HDB = XIDY*Q
      HDC = XIDZ*R
C
      PD = (TL- HDA - HCA)/XIX
      QD = (TM- HDB - HCB)/XIY
      RD = (TN- HDC - HCC)/XIZ
      GOTO 410
  402 PHID = PSID*STH
      UD = FXT/XMASS-W*Q+V*R+GRAV*A(1,3)
      VD = FYT/XMASS - U*R + GRAV*A(2,3)
      WD = FZT/XMASS + U*Q + GRAV*A(3,3)
C
      HCA = 0.
      HCB = 0.
      HCC = 0.
C
      HDA = P*XIDX
      HDB = Q*XIDY
      HDC = R*XIDY
C
      PD = (TL- HCA - HDA)/XIX
      QD = (TM- HCB - HDB)/XIY
      RD = (TN- HCC - HDC)/XIY
      GOTO 410
C
  403 PHID = P + PSID*STH
      UD = FXT/XMASS -W*Q + V*R + GRAV*A(1,3)
      VD = FYT/XMASS +W*P - U*R + GRAV*A(2,3)
      WD = FZT/XMASS + U*Q - V*P + GRAV*A(3.3)
C
      HA = P*XIX - Q*XIXY - R*XIXZ
      HB =-P*XIXY + Q*XIY  - R*XIYZ
      HC =-P*XIXZ - Q*XIYZ + R*XIZ
C
      HCA = Q*HC - R*HB
      HCB = R*HA - P*HC
      HCC = P*HB - Q*HA
C
      HDA = P*XIDX  - Q*XIDXY - R*XIDXZ
```

```
      HDB =-P*XIDXY + Q*XIDY  - R*XIDYZ
      HDC =-P*XIDXZ - Q*XIDYZ + R*XIDZ
C
      OMA = TL - HDA - HCA
      OMB = TM - HDB - HCB
      OMC = TN - HDC - HCC
C
      PD = OMA*YIX  + OMB*YIXY + OMC*YIXZ
      QD = OMA*YIXY + OMB*YIY  + OMC*YIYZ
      RD = OMA*YIXZ + OMC*YIYZ + OMC*YIZ
      GOTO 410
  410 CONTINUE
C
C     KINEMATIC RESTRICTIONS WHILE BIRD IS ON LAUNCHER
      S = SQRT((XI-XIO)**2+(YI-YIO)**2+(ZI)**2)
      IF(S-SGUID) 202,202,230
  202 QD = 0.0
      RD = 0.0
      PD = 0.0
      V = VO
      W = WO
      VD = 0.0
      UD = AMAX1(UD,0. )
      WD = 0.0
230   CONTINUE
      IF(KUTTA-4) 1,5,5
    5 KUTTA = 0
C
C     CALCULATION OF TRAJECTORY-PARAMETERS
C
      IF(QBAR-QBMAX(NSHOT)) 255,255,254
 254  QBMAX(NSHOT) = QBAR
  255 IF(XMACH-XMMAX(NSHOT)) 257,257,256
  256 XMMAX(NSHOT) = XMACH
  257 IF(ALT-APOGE(NSHOT)) 259,259,258
  258 APOGE(NSHOT) = ALT
      TAPOG(NSHOT) = TIME
  259 CONTINUE
C
C     CALCULATION OF DELT
C
      IF(LOMO) 231,231,232
  231 DELT = DEL1
      GOTO 233
  232 DELT = DEL2
  233 CONTINUE
C
C     END OF RUNGE-KUTTA LOOP
C     END OF SHOT?
C
      IF(ZDI) 240,240,241
  241 IF(-ZI-ALTST-ZDI*DELT*0.5) 290,290,240
  240 IF(TIME-TSTOP+DELT*0.5) 250,290,290
C
C     OUTPUT WANTED?
C
  250 IF(TIME-PTIME+.001*DELT) 1,251,251
  251 IOUT = IOUT+1
      PTIME = PTIME+DELPT
      IF(IOUT-199) 252,290,290
C
```

```
C     OUTPUT ARRAY IS FILLED
C
  252 O(1,IOUT) = TIME
      O(2,IOUT) = XI
      O(3,IOUT) = YI
      O(4,IOUT) = - ZI
C
C  -ZI = ALTITUDE CENTRE-OF-MASS ABOVE GROUND LEVEL.
C
      O(5,IOUT) = SQRT(U**2+V**2+W**2)
      TEMP = SQRT(XDI**2+YDI**2)
      O(6,IOUT) = CNVRG * ATAN2(-ZDI,TEMP)
      O(7,IOUT) = CNVRG * ATAN2(YDI,XDI)
      O(8,IOUT) = UD
      O(9,IOUT) = VD
      O(10,IOUT) = WD
      O(11,IOUT) = THET*CNVRG
      O(12,IOUT) = PSI *CNVRG
      O(13,IOUT) = PHI *CNVRG
      O(14,IOUT) = P   *CNVRG
      O(15,IOUT) = Q   *CNVRG
      O(16,IOUT) = R   *CNVRG
      O(17,IOUT) = PD * CNVRG
      O(18,IOUT) = QD  *CNVRG
      O(19,IOUT) = RD  *CNVRG
      O(20,IOUT) =ALPHA*CNVRG
      O(21,IOUT) = BETA*CNVRG
      O(22,IOUT) = QBAR
      O(23,IOUT) = XMACH
      O(24,IOUT) = FAX
      O(25,IOUT) = FAY
      O(26,IOUT) = FAZ
      O(27,IOUT) = FEX
      O(28,IOUT) = FEY
      O(29,IOUT) = FEZ
      O(30,IOUT) = AL
      O(31,IOUT) = AM
      O(32,IOUT) = AN
      O(33,IOUT) = EML
      O(34,IOUT) = EMM
      O(35,IOUT) = EMN
      O(36,IOUT) = WGHT
      O(37,IOUT) = XCGT
      O(38,IOUT) = XCP
      O(39,IOUT) = XIY
      O(40,IOUT) = XIX
      O(41,IOUT) = YCGT
      O(42,IOUT) = ZCGT
      O(43,IOUT) = XIZ
      O(44,IOUT) = XIXY
      O(45,IOUT) = XIXZ
      O(46,IOUT) = XIYZ
      O(47,IOUT) =-HCA
      O(48,IOUT) =-HCB
      O(49,IOUT) =-HCC
      O(50,IOUT) =-HDA
      O(51,IOUT) =-HDB
      O(52,IOUT) =-HDC
      O(53,IOUT) = WX
      O(54,IOUT) = WY
      O(55,IOUT) = WZ
```

```
      WW1=WX*A(1,1)+WY*A(2,1)+WZ*A(3,1)
      WW2=WX*A(1,2)+WY*A(2,2)+WZ*A(3,2)
      WW3=WX*A(1,3)+WY*A(2,3)+WZ*A(3,3)
      O(56,IOUT) = WW1
      O(57,IOUT) = WW2
      O(58,IOUT) = WW3
      O(59,IOUT) = CNVRG*ATAN(-ZI/(18500./.3048-XI))
      O(60,IOUT) = CNVRG*ATAN(-YI/(18500./.3048-XI))
C
C     OPTIONAL OUTPUT ON-LINE FOR NCOUT.GE.0
C
      IF(NCOUT) 269,261,261
  261 IF(NCOUT-10) 263,262,262
  262 OO(1) = XI*.3048
C
      OO(2) = -ZI*.3048
      OO(3) = O(5,IOUT)*.3048
      OO(4) = QBAR*47.9
      OO(5) = FEX*4.45
      OO(6) = FAX*4.45
      OO(7) = WGHT*.453
      WRITE(NOUT,110) TIME,(OO(I),I=1,3),O(6,IOUT),O(20,IOUT),O(14,IOUT)
     *,XMACH,(OO(I),I=4,7)
      GOTO 269
  263 WRITE(NOUT,110) TIME,XI,ALT,O(5,IOUT),O(6,IOUT),O(20,IOUT),
     *O(14,IOUT),XMACH,QBAR,FEX,FAX,WGHT
  269 CONTINUE
C
      IF(KILL) 1,1,299
  290 KILL = 1
      IOUT = IOUT + 1
      GOTO 252
C
C     END OF SHOT
C
  299 ICOUT = NCOUT
      CALL OUTPU(O,IOUT,NOUT,ICOUT,NSHOT)
      RANGE(NSHOT) = XI
      FLTIM(NSHOT) = TIME
      CALL OUTAB(NOUT,NCOUT,NSHOT,RANGE,FLTIM,APOGE,TAPOG,QBMAX,XMMAX,1)
  999 CONTINUE
C
C     END OF RUN
C
      CALL OUTAB(NOUT,NCOUT,NFIN ,RANGE,FLTIM,APOGE,TAPOG,QBMAX,XMMAX,2)
      STOP
  100 FORMAT(1H1,X,40(1H*),' RESULTS SHOT NO. ',I3,X,40(1H*),//,
     *3X,'TIME',6X,'RANGE',X,'ALTITUDE',X,'VELOCITY',4X,'ELEV',4X,
     *'ALPHA',X,'ROLLRATE',5X,'MACH',X,'DYN.PRES',3X,'THRUST',5X,'DRAG',
     *4X,'WEIGHT')
  101 FORMAT(3X,'(SEC)',4X,'(M)',5X,'(M)',5X,'(M/S)',5X,'(DEG)',4X,
     *'(DEG)',2X,'(DEG/S)',5X,'( )',3X,'(N/M2)',5X,'(N)',7X,'(N)',5X,
     *'(KG)',/)
  102 FORMAT(3X,'(SEC)',3X,'(FT)',4X,'(FT)',4X,'(FT/S)',5X,'(DEG)',4X,
     *'(DEG)',2X,'(DEG/S)',5X,'( )',2X,'(LB/FT2)',4X,'(LB)',6X,'(LB)',
     *4X,'(LB)',/)
  110 FORMAT(F8.2,F9.0,F9.0,F9.1,F9.2,F9.2,F9.0,F9.2,F9.0,F9.1,F9.1,F9.2
     *)
      END
```

```
      SUBROUTINE INPUS(NIN,NOUT,NSHOT,O,IOUT)
      COMMON /INDAT/ NOROL,DEL1,TSTOP,ALTST,DELPT,NCOUT,NIOUT,DEL2,
     *SLAUN,SQUID,SFRLN,XIO,YIO,ALTO,UO,VO,WO,THETO,PSIO,PHIO,PO,QO,RO,
     *CLP,CMQ,CLDA,DELTF,B,NMACH,AMACH(20),CDOT(20),CNAT(20),XCPT(20),
     *WTAPE,XCGTP,RCGTP,PCGTP,XIXTP,XIYTP,DIYZ,WPLD,XCGPL,RCGPL,PCGPL,
     *CLFIN,CPFIN,DDFIN,PHFIN,FIFAC,DXCP,FXCN,FXCD,
     *SREF,
     *MTYPE,WTE(5),WTPR(5),SPIMP(5),CTHR(5),CTHRT(5),NTTHR(5),
     *TTHRT(100,5),TTHR(100,5),COR1(5),COR2(5),
     *NKICK,NTYPE(10),NMOT(10),XTHR(10),ATHR(10),DTHR(10),PTHR(10),
     *XEMPT(10),REMPT(10),PEMPT(10),XPROP(10),RPROP(10),PPROP(10),
     *TIG(10),RTHR(10),TORQL(10),TCCD(10),TCCL(10),TCCP(10),
     *NALTW,ALTW(100),WIXA(100),WIYA(100),WIZA(100),
     *NALTA,ALTA(100),TMPR(100),PRES(100),
     * NFIN
      DIMENSION O(60,200)
      IF(NSHOT.EQ.1) THETO=45.0
      IF(NSHOT.EQ.2) THETO=50.0
      IF(NSHOT.EQ.3) THETO=55.0
      IF(NSHOT.EQ.4) THETO=60.0
      IF(NSHOT.EQ.5) THETO=65.0
      IF(NSHOT.EQ.6) THETO=70.0
      WRITE(6,1) NSHOT,THETO
1     FORMAT(1H1,'CASE= ',I5,'THETO(DEG) = ',1PG15.5)
      RETURN
      END
```

```
      SUBROUTINE INPUT (NIN,NOUT)
C
      COMMON /INDAT/ NOROL,DEL1,TSTOP,ALTST,DELPT,NCOUT,NIOUT,DEL2,
     *SLAUN,SGUID,SFRLN,XIO,YIO,ALTO,UO,VO,WO,THETO,PSIO,PHIO,PO,QO,RO,
     *CLP,CMQ,CLDA,DELTF,B,NMACH,AMACH(20),CDOT(20),CNAT(20),XCPT(20),
     *WTAPE,XCGTP,RCGTP,PCGTP,XIXTP,XIYTP,DIYZ,WPLD,XCGPL,RCGPL,PCGPL,
     *CLFIN,CPFIN,DDFIN,PHFIN,FIFAC,DXCP,FXCN,FXCD,
     *SREF,
     *MTYPE,WTE(5),WTPR(5),SPIMP(5),CTHR(5),CTHRT(5),NTTHR(5),
     *TTHRT(100,5),TTHR(100,5),COR1(5),COR2(5),
     *NKICK,NTYPE(10),NMOT(10),XTHR(10),ATHR(10),DTHR(10),PTHR(10),
     *XEMPT(10),REMPT(10),PEMPT(10),XPROP(10),RPROP(10),PPROP(10),
     *TIG(10),RTHR(10),TORQL(10),TCCD(10),TCCL(10),TCCP(10),
     *NALTW,ALTW(100),WIXA(100),WIYA(100),WIZA(100),
     *NALTA,ALTA(100),TMPR(100),PRES(100),
     * NFIN
C
C
C
      READ(NIN,100) NOROL,NCOUT,NIOUT
      READ(NIN,101) DEL1,DEL2,TSTOP,ALTST,DELPT
      READ(NIN,101) SGUID,SFRLN
      READ(NIN,101) XIO,YIO,ALTO
      READ(NIN,101) UO,VO,WO
      READ(NIN,101) THETO,PSIO,PHIO
      READ(NIN,101) PO,QO,RO
      READ(NIN,101) CLP,CMQ,CLDA,DELTF,B,SREF
      READ(NIN,101) CLFIN,DDFIN,CPFIN,PHFIN,FIFAC
      READ(NIN,101) DXCP,FXCN,FXCD
C
      READ(NIN,100) NMACH
      DO 1 I=1,NMACH
    1 READ(NIN,101) AMACH(I),CDOT(I),CNAT(I),XCPT(I)
C
      READ(NIN,101) WTAPE,XCGTP,RCGTP,PCGTP,XIXTP,XIYTP,DIYZ
      READ(NIN,101) WPLD,XCGPL,RCGPL,PCGPL
C
      READ(NIN,100) MTYPE
      DO 2 I=1,MTYPE
      READ(NIN,101) WTE(I),WTPR(I),SPIMP(I),CTHR(I),CTHRT(I)
      READ(NIN,100) NTTHR(I)
      J=NTTHR(I)
      DO 3 K = 1,J
    3 READ (NIN,101) TTHRT(K,I),TTHR(K,I)
    2 CONTINUE
C
      READ(NIN,100) NKICK
      DO 4 I=1,NKICK
      READ(NIN,100) NTYPE(I),NMOT(I)
      READ(NIN,101) TIG(I),XTHR(I),RTHR(I),PTHR(I),DTHR(I),ATHR(I)
      READ(NIN,101) XEMPT(I),XPROP(I),REMPT(I),RPROP(I),PEMPT(I),
     *PPROP(I)
    4 READ(NIN,101) TORQL(I),TCCD(I),TCCL(I),TCCP(I)
C
      READ(NIN,100) NALTW
      IF(NALTW) 5,5,6
    6 DO 7 I=1,NALTW
    7 READ (NIN,101) ALTW(I),WIXA(I),WIYA(I),WIZA(I)
C
    5 READ(NIN,100) NALTA
      IF(NALTA) 8,8,9
```

```
    9 DO 10 I=1,NALTA
   10 READ (NIN,101) ALTA(I),TMPR(I),PRES(I)
C
    8 CONTINUE
      READ(NIN,100) NFIN
      IF(NIOUT) 99,99,11
   11 WRITE(NOUT,200)
      WRITE(NOUT,201) NOROL,NCOUT,NIOUT
      WRITE(NOUT,202) DEL1,DEL2,TSTOP,ALTST,DELPT
      WRITE(NOUT,203) SGUID,SFRLN
      WRITE(NOUT,204) XIO,YIO,ALTO
      WRITE(NOUT,205) UO,VO,WO
      WRITE(NOUT,206) THETO,PSIO,PHIO
      WRITE(NOUT,239) PO,QO,RO
      WRITE(NOUT,207) CLP,CMQ,CLDA,DELTF,B,SREF
      WRITE(NOUT,221) CLFIN,DDFIN,CPFIN,PHFIN,FIFAC
      WRITE(NOUT,222) DXCP,FXCN,FXCD
C
      WRITE(NOUT,208) NMACH
      DO 12 I=1,NMACH
   12 WRITE (NOUT,209) AMACH(I),CDOT(I),CNAT(I),XCPT(I)
C
      WRITE(NOUT,210) WTAPE,XCGTP,RCGTP,PCGTP,XIXTP,XIYTP,DIYZ
     *,WPLD,XCGPL,RCGPL,PCGPL
C
      WRITE(NOUT,211) MTYPE
      DO 13 I=1,MTYPE
      WRITE(NOUT,212) WTE(I),WTPR(I),SPIMP(I),NTTHR(I),CTHR(I),CTHRT(I)
      J=NTTHR(I)
      DO 14 K=1,J
   14 WRITE(NOUT,213) TTHRT(K,I),TTHR(K,I)
   13 CONTINUE
C
      WRITE(NOUT,214) NKICK
      DO 15 I=1,NKICK
   15 WRITE(NOUT,215) NTYPE(I),NMOT(I),TIG(I),XTHR(I),RTHR(I),PTHR(I),
     *DTHR(I),ATHR(I),
     *XEMPT(I),REMPT(I),PEMPT(I),XPROP(I),RPROP(I),PPROP(I),
     *TORQL(I),TCCD(I),TCCL(I),TCCP(I)
C
      WRITE(NOUT,216) NALTW
      IF(NALTW) 16,16,17
   17 DO 18 I=1,NALTW
   18 WRITE (NOUT,217) ALTW(I),WIXA(I),WIYA(I),WIZA(I)
C
   16 WRITE (NOUT,218) NALTA
      IF(NALTA) 19,19,20
   20 DO 21 I=1,NALTA
   21 WRITE(NOUT,219) ALTA(I),TMPR(I),PRES(I)
C
   19 CONTINUE
      WRITE(NOUT,220) NFIN
   99 CONTINUE
      RETURN
C
C
C
  100 FORMAT(7I2)
  101 FORMAT(8F10.2)
  102 FORMAT(20I3)
  103 FORMAT(18A4)
```

```
  104 FORMAT(3A4,/,3A4,/,3A4)
C
  200 FORMAT(30X,'INPUT DATA',/,X,100(1H*),/)
  201 FORMAT(X,'NOROL,NCOUT,NIOUT',3I5)
  202 FORMAT(X,'DEL1,DEL2,TSTOP,ALTST,DELPT',2F10.7,4F10.2)
  203 FORMAT(X,'SGUID,SFRLN                    ',3F10.2)
  204 FORMAT(X,'XIO,YIO,ALTO                   ',3F10.2)
  205 FORMAT(X,'UO,VO,WO                       ',3F10.2)
  206 FORMAT(X,'THETO,PSIO,PHIO                ',3F10.2)
  207 FORMAT(X,'CLP,CMQ,CLDA,DELTF,B,SREF      ',6F10.2,/)
  221 FORMAT(X,'CLFIN,DDFIN,CPFIN,PHFIN,FIF',5F10.2,/)
  222 FORMAT(X,'DXCP,FXCN,FXCD                 ',3F10.2,/)
  208 FORMAT(X,'AERODYNAMICS NMACH=',I2,
     */,'      AMACH       CDOT       CNAT       XCPT',/)
  209 FORMAT(X,F9.2,3F10.4)
  210 FORMAT(/,X,'WTAPE,XCGTP,RCGTP,PCGTP,XIXTP,XIYTP,DIYZ=',
     *7F10.2,/,X,'WPLD,XCGPL,RCGPL,PCGPL=                 ',
     *4F10.2,/)
  211 FORMAT(X,'NO.OF MOTOR-TYPES MTYPE=',I2)
  212 FORMAT(/,X,'WTE,WTPR,SPIMP',3F10.2,/,
     *          X,'NTHR,CTHR,CTHRT          ',I10,2F10.2,
     */,X,'THRUST',
     */,'      TTHRT     THRT')
  213 FORMAT(F10.4,F10.1)
  214 FORMAT(/,X,'NO.OF KICKS NKICK=',I2,
     */,' NTYPE NMOT    TIG    XTHR    RTHR    PTHR    DTHR    ATHR    XEMPT
     *  REMPT  PEMPT  XPROP  RPROP  PPROP  TORQL   TCCD    TCCL    TCCP',/)
  215 FORMAT(2I5,3X,12F7.2,F7.5,F7.4,3F7.2)
  216 FORMAT(/,X,'WIND NALTW=',I3,
     */,'       ALTW       WIXA       WIYA       WIZA',/)
  217 FORMAT(F10.0,3F10.1)
  218 FORMAT(/,X,'ATMOSPHERIC DATA NALTA=',I3,
     */,'       ALTA       TEMP       PRES',/)
  219 FORMAT(F10.0,F10.2,F10.3)
  220 FORMAT(/,X,'NO.OF SHOTS NFIN=',I3)
  239 FORMAT(X,'PO,QO,RO                       ',3F10.2,/)
C
C
C
      END
```

```
      SUBROUTINE RUNK1(KUTTA,DT,X,XD,I)
      DIMENSION C1(25),C2(25),C3(25),C4(25),SX(25)
      COMMON/DEQ/C1,C2,C3,C4,SX
      GO TO (1,2,3,4),KUTTA
    1 SX(I)=X
      C1(I)=XD*DT
      X=SX(I)+0.5*C1(I)
      RETURN
    2 C2(I)=XD*DT
      X = SX(I)+0.5*C2(I)
      RETURN
  3   C3(I) = XD*DT
      X = SX(I)+C3(I)
      RETURN
  4   C4(I) = XD*DT
      X = SX(I)+(C1(I)+C4(I)+2.*(C2(I)+C3(I)))/6.
      RETURN
      END
```

```
      SUBROUTINE ATMOS(RHO,PP,TK,VS,G,ALT,TT,TP,TA,N,RHFCT,VSFCT)
      DIMENSION TT(100),TP(100),TD(100),TA(100)
      DATA RE /20855531./
      R = RE/(ALT+RE)
      H = .3048*ALT*R
      TK = 288.15-0.0065*H
      PN = 101300.*(TK/288.15)**(+5.255)
      IF(N.LE.1) GOTO 3
      CALL INTEG(TT,TP,TD,TA,DT,DP,D,ALT,N)
      TK = TK*(1.+DT/100.)
      PN = PN*(1.+DT/100.)
    3 CONTINUE
      PP = 0.020885*PN
      RHO = 6.75976E-6*PN/TK
      VS = 65.7688*SQRT(TK)
      G = 32.1741*R**2
      RHO = RHO*RHFCT
      VS = VS*VSFCT
      RETURN
      END
```

```
      SUBROUTINE INTEQ (TY1,TY2,TY3,TX,Y1,Y2,Y3,X,N)
      N1=N-1
      IF(X.LT.TX(1)) GO TO 3
      IF(X.GT.TX(N)) GO TO 4
      DO 1 I=1,N1
      J=I
      IF((X-TX(I))*(X-TX(I+1))) 2,2,1
1     CONTINUE
4     CONTINUE
      J=N1
      GO TO 2
3     J=1
2     CONTINUE
      DTY1=TY1(J+1)-TY1(J)
      DTY2=TY2(J+1)-TY2(J)
      DTY3=TY3(J+1)-TY3(J)
      DX=(X-TX(J))/(TX(J+1)-TX(J))
      Y1=TY1(J)+DTY1*DX
      Y2=TY2(J)+DTY2*DX
      Y3=TY3(J)+DTY3*DX
      RETURN
      END
```

```
      SUBROUTINE WIND(TX,TY,TZ,TA,N,ALT,A,R1,R2,R3)
C
      DIMENSION TX(N),TY(N),TZ(N),TA(N),A(3,3),R(3),X(3)
C
      CALL INTEQ(TX,TY,TZ,TA,X(1),X(2),X(3),ALT,N)
C
      DO 1 I=1,3
      R(I)=0
      DO 1 J=1,3
      R(I)=R(I)+X(J)*A(I,J)
1     CONTINUE
      R1=R(1)
      R2=R(2)
      R3=R(3)
      RETURN
      END
```

```
      SUBROUTINE INTPL (TY,TX,Y,X,K,N)
C
      DIMENSION TX(N),TY(N)
C
      N1 = N- 1
      DO 1 I = K,N1
      J=I
      IF((X-TX(I))*(X-TX(I+1))-0.0001) 2,2,1
    1 CONTINUE
      J=N1
    2 CONTINUE
      DTY = TY(J+1) - TY(J)
      Y = TY(J) + DTY*(X-TX(J))/(TX(J+1)-TX(J))
      K=J
      RETURN
      END
```

```
      SUBROUTINE OUTPU (A,IOUT,NOUT,NCOUT,NSHOT)
C
      DIMENSION A(60,200)
      DATA S,P,F /1.354,.453,.3048/
      NCONV=NCOUT/10
      NTABL=NCOUT-NCONV*10
      IF(NTABL-1) 99,1,1
    1 CONTINUE
C
C     1ST TABLE
C
      WRITE(NOUT,100) NSHOT
      IF(NCONV) 3,3,2
    3 WRITE(NOUT,101)
      GOTO 4
    2 WRITE(NOUT,102)
      DO 8 I=1,IOUT
      DO 7 J=2,5
    7 A(J,I)=A(J,I)*F
      DO 6 J=8,10
    6 A(J,I)=A(J,I)*F
    8 CONTINUE
    4 CONTINUE
      DO 5 I=1,IOUT
    5 WRITE(NOUT,103) (A(J,I),J=1,8),A(59,I),A(60,I)
C
      IF(NTABL-2) 99,10,10
C
C     2ND TABLE
C
   10 WRITE (NOUT,100) NSHOT
      WRITE(NOUT,110)
      DO 11 I=1,IOUT
   11 WRITE(NOUT,111) A(1,I),(A(J,I),J=11,19)
C
      IF(NTABL-3) 99,20,20
C
C     3RD TABLE
C
   20 WRITE(NOUT,100) NSHOT
      WRITE(NOUT,120)
      IF(NCONV) 21,21,22
   21 WRITE(NOUT,121)
      GOTO 24
   22 WRITE(NOUT,122)
      DO 27 I=1,IOUT
      A(22,I)=A(22,I)*P*9.8065/F**2
      A(36,I)=A(36,I)*P
      A(37,I)=A(37,I)*F
      A(38,I)=A(38,I)*F
      DO 27 J=53,55
   27 A(J,I) = A(J,I)*F
   24 DO 28 I=1,IOUT
   28 WRITE(NOUT,123) A(1,I),(A(J,I),J=20,23),(A(J,I),J=36,38),
     *(A(J,I),J=53,55)
C
      IF(NTABL-4) 99,30,30
C
C     4TH TABLE
C
   30 WRITE(NOUT,100) NSHOT
```

```
      WRITE(NOUT,130)
      IF(NCONV) 31,31,32
   31 WRITE(NOUT,131)
      GOTO 36
   32 WRITE(NOUT,132)
      DO 33 I=1,IOUT
      DO 34 J=24,35
   34 A(J,I)=A(J,I)*P*9.8065
      DO 35 J=30,35
   35 A(J,I)=A(J,I)*F
   33 CONTINUE
   36 DO 37 I=1,IOUT
   37 WRITE(NOUT,133) A(1,I),(A(J,I),J=24,35)
C
      IF(NTABL-5) 99,40,40
C
C     5TH TABLE
C
   40 WRITE(NOUT,100) NSHOT
      WRITE(NOUT,140)
      IF(NCONV) 41,41,42
   41 WRITE(NOUT,141)
      GOTO 47
   42 WRITE(NOUT,142)
      DO 43 I=1,IOUT
      DO 45 J = 56,58
   45 A(J,I) = A(J,I) * F
      DO 46 J=47,52
   46 A(J,I) = A(J,I) * P*F*9.8065
   43 CONTINUE
   47 DO 48 I = 1,IOUT
   48 WRITE(NOUT,143) A(1,I),(A(J,I),J=47,52),(A(J,I),J=56,58)
C
      IF(NTABL-6) 99,50,50
C
C     6TH TABLE
C
   50 WRITE(NOUT,100) NSHOT
      WRITE(NOUT,150)
      IF(NCONV) 51,51,52
   51 WRITE(NOUT,151)
      GOTO 58
   52 WRITE(NOUT,152)
      DO 53 I=1,IOUT
      A(41,I) = A(41,I) * F
      A(42,I) = A(42,I) * F
      A(39,I) = A(39,I) * S
      A(40,I) = A(40,I) * S
      DO 54 J=43,46
   54 A(J,I) = A(J,I) * S
   53 CONTINUE
   58 DO 59 I=1,IOUT
   59 WRITE(NOUT,153) A(1,I),(A(J,I),J=36,37),A(41,I),A(42,I),
     *A(40,I),A(39,I),(A(J,I),J=43,46)
C
C
C
   99 CONTINUE
      RETURN
  100 FORMAT(1H1,/,X,40(1H*),' RESULTS SHOT NO. ',I2,X,40(1H*),//)
  101 FORMAT ('     TIME       RANGE      CROSS-R        ALT       VELOC        ELE
```

```
     *V         AZIM       U-DOT      ASP-ELV    ASP-AZ',/,
     *            '        (SEC)      (FT)       (FT)       (FT)       (FT/S)     (DE
     *G)        (DEG)      (FT/S2)    (DEG)      (DEG)  ',/)
  102 FORMAT ('       TIME       RANGE      CROSS-R      ALT       VELOC      ELE
     *V         AZIM       U-DOT      ASP-ELEV   ASP-AZ',/,
     *            '        (SEC)      (M)        (M)        (M)        (M/S)      (DE
     *G)        (DEG)      (M/S2)     (DEG)      (DEG)',/)
  103 FORMAT (F10.2,3F10.0,6F10.2)
  110 FORMAT ('         TIME      THETA      'PSI         PHI        P          Q
     *           R          P-DOT      Q-DOT      RDOT',/,
     *            '         (SEC)     (DEG)      (DEG)      (DEG)      (DEG/S)    (DEG/
     *S)      (DEG/S)    (DEG/S2)   (DEG/S2)   (DEG/S2)',/)
  111 FORMAT (10F10.2)
  120 FORMAT ('       TIME      ALPHA       BETA     DYN.PRESS     MACH       WGH
     *T         C.G        C.P        WIND-X     WIND-Y     WIND-Z')
  121 FORMAT ('       (SEC)      (DEG)      (DEG)    (LB/FT2)       (-)       (LB
     *)         (FT)       (FT)       (FT/S)     (FT/S)     (FT/S)',/)
  122 FORMAT ('       (SEC)      (DEG)      (DEG)    (N/M2)         (-)       (KG
     *)         (M)        (M)        (M/S)      (M/S)      (M/S)',/)
  123 FORMAT (11F10.2)
  130 FORMAT('      TIME',
     *            '  FX-AERO   FY-AERO   FZ-AERO   FX-THR    FY-THR    FZ-THR    TX-A
     *ERO    TY-AERO   TZ-AERO   TX-THR    TY-THR    TZ-THR')
  131 FORMAT('     (SEC)',
     *            '    (LB)      (LB)      (LB)      (LB)      (LB)      (LB)      (LB*
     *FT)    (LB*FT)   (LB*FT)   (LB*FT)   (LB*FT)   (LB*FT)',/)
  132 FORMAT('     (SEC)',
     *            '    (N)       (N)       (N)       (N)       (N)       (N)       (N*
     *M)       (N*M)     (N*M)     (N*M)     (N*M)     (N*M)',/)
  133 FORMAT(F9.2,F9.0,2F9.2,F8.0,2F8.1,3F9.1,3F8.1)
  140 FORMAT(       '          TIME       COUP-L     COUP-M     COUP-N     DAMP-L ',
     *    '    DAMP-M     DAMP-N     WW1        WW2        WW3      ')
  141 FORMAT(       '          (SEC)      (LB*FT)    (LB*FT)    (LB*FT)    (LB*FT) ',
     *    '  (LB*FT)    (LB*FT)    (FPS)      (FPS)      (FPS)    ',/)
  142 FORMAT(       '          (SEC)      (N*M)      (N*M)      (N*M)      (N*M)   ',
     *    '    (N*M)      (N*M)      (M/S)      (M/S)      (M/S)',/)
  143 FORMAT(10F10.2)
  150 FORMAT(       '          TIME       WEIGHT     X-CG       Y-CG       Z-CG    ',
     *    '     I-XX       I-YY       I-ZZ       I-XY       I-XZ       I-YZ',/)
  151 FORMAT(       '          (SEC)      (LB)       (FT)       (FT)       (FT)    ',
     *    '  (SL*FT2)   (SL*FT2)   (SL*FT2)(SL*FT2)   (SL*FT2)   (SL*FT2)',
     */)
  152 FORMAT(       '          (SEC)      (KG)       (M)        (M)        (M)     ',
     *    '  (KG*M2)    (KG*M2)    (KG*M2)    (KG*M2)    (KG*M2)    (KG*M2)',/
     *)
  153 FORMAT(11F10.2)
C
C
C
      END
```

```
      SUBROUTINE OUTAB (NOUT,NCOUT,NSHOT,RANGE,FLTIM,APOGE,TAPOG,QBMAX,
     *XMMAX,K)
      DIMENSION RANGE(100),FLTIM(100),APOGE(100),TAPOG(100),QBMAX(100),
     *XMMAX(100)
      DATA F,P /.3048,.453/
C
      GOTO (1,2),K
    1 I=NSHOT
      WRITE(NOUT,100) I,RANGE(I),FLTIM(I),APOGE(I),TAPOG(I),XMMAX(I),
     *QBMAX(I)
      RETURN
    2 WRITE(NOUT,110)
      IF(NCOUT-10) 3,3,4
    4 WRITE(NOUT,114)
      DO 5 I=1,NSHOT
      RANGE(I) = RANGE(I)*F
      APOGE(I) = APOGE(I)*F
    5 QBMAX(I) = QBMAX(I)*P*9.8065/F**2
      GOTO 6
    3 WRITE(NOUT,115)
    6 DO 7 I=1,NSHOT
    7 WRITE(NOUT,120) I,RANGE(I),APOGE(I),FLTIM(I),TAPOG(I),XMMAX(I),
     *QBMAX(I)
      RETURN
C
C
C
  100 FORMAT(///,X,10(1H*),' SUMMARY:NO,RANGE,FLIGHTTIME,APOGEE,T-APOGEE
     *,MAX.MACH,MAX.DYN.PRESS',/,I22,F8.0,F8.1,F8.0,F8.1,F8.2,F14.1)
  110 FORMAT(1H1,X,40(1H*),' SUMMARY-TABLE ',40(1H*),//,X,
     *'NO.',5X,'RANGE',4X,'APOGEE',X,'FLIGHT-TIME',X,'T-APOGEE',X,
     *'MAX.MACH',X,'MAX.DYN.PRES')
  114 FORMAT(10X,'(M)',6X,'(M)',6X,'(SEC)',5X,'(SEC)',5X,'( )',7X,'(N/M2
     *)',/)
  115 FORMAT(10X,'(FT)',5X,'(FT)',5X,'(SEC)',5X,'(SEC)',5X,'( )',6X,'(LB
     */FT2)',/)
  120 FORMAT(I3,F11.0,F9.0,F11.2,F10.2,F7.2,F13.1)
      END
```

APPENDIX 2

THRUST FORCES AND MOMENTS RESOLVED IN F_B

APPENDIX 2

THRUST FORCE AND MOMENTS RESOLVED IN F_B

In Section 2.6 of the main text, the thrust forces and moments were presented in terms of components in the body-fixed reference frame F_B. In order to obtain these values, however, a transformation is required taking the thrust vector and transforming it into components in F_B and then using these components and the location of the point of action of the thrust vector relative to F_B to compute the thrust moments. This Appendix summarizes this analysis.

Only the case for one rocket motor of arbitrary orientation relative to the fuselage reference line (FRL) is considered. The results for several rocket motors follow with an algebraic summation of each motor's contributions.

A reference frame F_{TH_i} is defined in Figure A2.1. F_{TH_i} has its origin located at the point of action on the airframe of the thrust vector of the i-th rocket motor, and its x-axis in the negative thrust direction. The orientation of the axes of F_{TH_i} relative to axes of the structural reference frame F_R is given by the ordered Euler angle rotations (ϕ_{TH_i}, θ_{TH_i}, δ_{TH_i}), i.e. a rotation ϕ_{TH_i} about the x-axis of F_R, a rotation θ_{TH_i} about the y-axis of the reference frame resulting from the first rotation, and a rotation δ_{TH_i} about the z-axis of the reference resulting from the second rotation. A further, important constraint is that the rotation ϕ_{TH_i} corresponds to the angular cylindrical coordinate of the point of action of the thrust vector of the i-th rocket motor relative to the FRL (see Figures 4 and A2.1).

In terms of the data inputs to the BALSIM package, the Euler angles (ϕ_{TH_i}, θ_{TH_i}, δ_{TH_i}) have a one-to-one correspondence with the program variables (PTHR(I), DTHR(I), ATHR(I)) (see Appendix 1).

Since the orientation of F_B is related to the orientation of F_R by a π rotation about the y-axis of F_R, it may be shown that the rotation matrix $\underline{L}_{BTH_i}$ relating components of a vector expressed in F_{TH_i} to the components of the same vector expressed in F_B is given by:

$$\underline{L}_{BTH_i} = \begin{bmatrix} -\cos\delta_{TH_i}\cos\theta_{TH_i} & \sin\delta_{TH_i}\cos\theta_{TH_i} & -\sin\theta_{TH_i} \\ \cos\delta_{TH_i}\sin\theta_{TH_i}\sin\phi_{TH_i} + \sin\delta_{TH_i}\cos\phi_{TH_i} & -\sin\delta_{TH_i}\sin\theta_{TH_i}\sin\phi_{TH_i} + \cos\delta_{TH_i}\cos\phi_{TH_i} & -\cos\theta_{TH_i}\sin\phi_{TH_i} \\ \cos\delta_{TH_i}\sin\theta_{TH_i}\cos\phi_{TH_i} - \sin\delta_{TH_i}\sin\phi_{TH_i} & -\sin\delta_{TH_i}\sin\theta_{TH_i}\cos\phi_{TH_i} - \cos\delta_{TH_i}\sin\phi_{TH_i} & -\cos\theta_{TH_i}\cos\phi_{TH_i} \end{bmatrix} \quad \text{(A2,1)}$$

From the definition of F_{TH_i} it follows that the thrust of the i-th rocket motor is given by:

$$\underline{T}_i^{TH_i} = (-T_i, 0, 0)^T \quad \text{(A2,2)}$$

and the nozzle moment of the i-th rocket motor is given by:

$$(\underline{M}_{T_i}^{TH_i})_{nz} = (-L_{nz_i}, 0, 0)^T \quad \text{(A2,3)}$$

Using (A2,1) and summing over all the rocket motors, it follows that:

$$X_{T_B} = \sum_{i=1}^{N_M} T_i \cos\delta_{TH_i}\cos\theta_{TH_i} \quad \text{(A2,4a)}$$

$$Y_{T_B} = \sum_{i=1}^{N_M} -T_i(\cos\delta_{TH_i}\sin\theta_{TH_i}\sin\phi_{TH_i} + \sin\delta_{TH_i}\cos\phi_{TH_i}) \quad \text{(A2,4b)}$$

$$Z_{T_B} = \sum_{i=1}^{N_M} -T_i(\cos\delta_{TH_i}\sin\theta_{TH_i}\cos\phi_{TH_i} - \sin\delta_{TH_i}\sin\phi_{TH_i}) \quad \text{(A2,4c)}$$

and

$$(L_{T_B})_{nz} = \sum_{i=1}^{N_M} L_{nz_i}\cos\delta_{TH_i}\cos\theta_{TH_i} \quad \text{(A2,5a)}$$

$$(M_{T_B})_{nz} = \sum_{i=1}^{N_M} -L_{nz_i}(\cos\delta_{TH_i}\sin\theta_{TH_i}\sin\phi_{TH_i} + \sin\delta_{TH_i}\cos\phi_{TH_i}) \quad \text{(A2,5b)}$$

$$(N_{T_B})_{nz} = \sum_{i=1}^{N_M} -L_{nz_i}(\cos\delta_{TH_i}\sin\theta_{TH_i}\cos\phi_{TH_i} - \sin\delta_{TH_i}\sin\phi_{TH_i}) \tag{A2,5c}$$

What remains is to compute the thrust moment $(\underset{\rightarrow}{M}_{Ti})_{cg}$ generated by the i-th rocket motor due to the displacement of its point of action relative to the vehicle centre-of-mass. Let $\underset{\rightarrow}{r}_{TH_i}$ be the vector from the vehicle centre-of-mass to the origin of F_{TH_i} (see Figure A2.1), $(x_{TH_i}, y_{TH_i}, z_{TH_i})$ are the coordinates of the origin of F_{TH_i}, in F_R, and (x_{cg}, y_{cg}, z_{cg}) are the coordinates of the vehicle centre-of-mass in F_R. $(\underset{\rightarrow}{M}_{Ti})_{cg}$ is given by:

$$(\underset{\rightarrow}{M}_{T_i})_{cg} = \underset{\rightarrow}{r}_{TH_i} \times \underset{\rightarrow}{T}_i \tag{A2,6}$$

or

$$\underline{M}^B_{T_i} = \underline{\tilde{r}}^B \, \underline{T}^B_i \tag{A2,7}$$

where $\tilde{r}^B$ is the matrix equivalent of the vector cross-product (Reference 3). Using the definitions of the previous paragraph and Figure A2.1, it may be shown that:

$$\underline{\tilde{r}}^B = \begin{bmatrix} 0 & (z_{TH_i} - z_{cg}) & (y_{TH_i} - y_{cg}) \\ -(z_{TH_i} - z_{cg}) & 0 & (x_{TH_i} - x_{cg}) \\ -(y_{TH_i} - y_{cg}) & -(x_{TH_i} - x_{cg}) & 0 \end{bmatrix} \tag{A2,8}$$

From (A2,4), (A2,7) and (A2,8), it follows that:

$$\begin{aligned}(L_{T_B})_{cg} = \sum_{i=1}^{N_M} [&-(z_{TH_i} - z_{cg})T_i(\cos\delta_{TH_i}\sin\theta_{TH_i}\sin\phi_{TH_i} + \sin\delta_{TH_i}\cos\phi_{TH_i}) \\ &-(y_{TH_i} - y_{cg})T_i(\cos\delta_{TH_i}\sin\theta_{TH_i}\cos\phi_{TH_i} - \sin\delta_{TH_i}\sin\phi_{TH_i})]\end{aligned} \tag{A2,9a}$$

$$\begin{aligned}(M_{T_B})_{cg} = \sum_{i=1}^{N_M} [&-(z_{TH_i} - z_{cg})T_i\cos\delta_{TH_i}\cos\theta_{TH_i} \\ &-(x_T - x_{cm})T_i(\cos\delta_{TH_i}\sin\theta_{TH_i}\cos\phi_{TH_i} - \sin\delta_{TH_i}\sin\phi_{TH_i})]\end{aligned} \tag{A2,9b}$$

$$(N_{T_B})_{cg} = \sum_{i=1}^{N_M} [-(y_{TH_i} - y_{cg}) T_i \cos\delta_{TH_i} \cos\theta_{TH_i}$$
$$+ (x_T - x_{cg}) T_i (\cos\delta_{TH_i} \sin\theta_{TH_i} \cos\phi_{TH_i} + \sin\delta_{TH_i} \cos\phi_{TH_i})] \quad \text{(A2,9c)}$$

All the terms that are required to specify the total thrust moments (L_{T_B}, M_{T_B}, N_{T_B}) as given by equations (2.6,1a), (2.6,1b) and (2.6,1c) have now been specified.

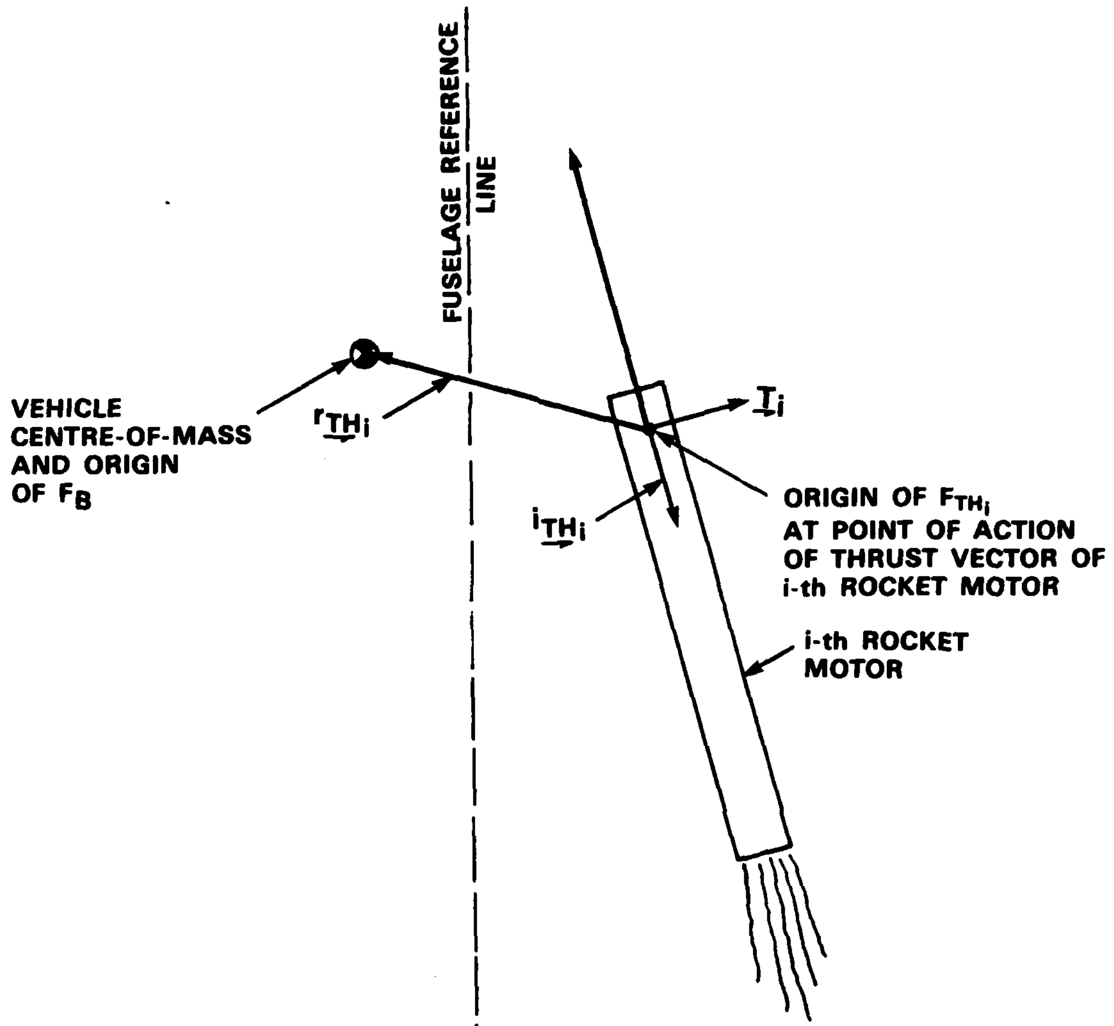

FIGURE A2.1

DEFINITION OF REFERENCE FRAME F_{TH_i}

UNCLASSIFIED

This Sheet Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1 ORIGINATING ACTIVITY Defence Research Establishment Suffield Ralston, Alberta	2a. DOCUMENT SECURITY CLASSIFICATION UNCLASSIFIED 2b. GROUP

3 DOCUMENT TITLE

A Nonlinear Six-Degree-Of-Freedom Ballistic Aerial Target Simulation Model (U)

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Suffield Memorandum

5. AUTHOR(S) (Last name, first name, middle initial)

Markov, A.B.

6. DOCUMENT DATE [redacted] August 1983	7a. TOTAL NO. OF PAGES 91	7b. NO. OF REFS 9

8a. PROJECT OR GRANT NO. 21V10	9a. ORIGINATOR'S DOCUMENT NUMBER(S) SM 1081
8b. CONTRACT NO.	9b. OTHER DOCUMENT NO.(S) (Any other numbers that may be assigned this document)

10. DISTRIBUTION STATEMENT

UNLIMITED

11. SUPPLEMENTARY NOTES	12. SPONSORING ACTIVITY DRES Ralston, Alberta

13. ABS. 'ACT

Six degree-of-freedom, rigid body equations of motion are described suitable for modeling the dynamic characteristics of multistaged, free-flight, ballistic rockets such as the DRES developed aerial targets CRV7/BATS and ROBOT-9. These equations of motion form the core of a FORTRAN simulation software package called BALSIM. This package allows for modeling of vehicle thrust and structural asymmetries, time-varying mass and inertia characteristics, variable wind conditions, nonstandard atmospheric conditions, stage failures, and different rocket motor types. The BALSIM package has been written in IBM FORTRAN IV and has been tested on the IBM 3033 computer with the H-extended compiler. It is currently being adapted for use with the VAX11/780 and Honeywell DPS-8/70C computers. (U)

UNCLASSIFIED

This Sheet Security Classification

KEY WORDS

Aerial Targets
CRV7/BATS
Flight Dynamics
ROBOT-9
Stability and Control
Six degree-of-freedom simulation

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the organization issuing the document.

2a. DOCUMENT SECURITY CLASSIFICATION: Enter the overall security classification of the document including special warning terms whenever applicable.

2b. GROUP: Enter security reclassification group number. The three groups are defined in Appendix 'M' of the DRB Security Regulations.

3. DOCUMENT TITLE: Enter the complete document title in all capital letters. Titles in all cases should be unclassified. If a sufficiently descriptive title cannot be selected without classification, show title classification with the usual one-capital-letter abbreviation in parentheses immediately following the title.

4. DESCRIPTIVE NOTES: Enter the category of document, e.g. technical report, technical note or technical letter. If appropriate, enter the type of document, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the document. Enter last name, first name, middle initial. If military, show rank. The name of the principal author is an absolute minimum requirement.

6. DOCUMENT DATE: Enter the date (month, year) of Establishment approval for publication of the document.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the document.

8a. PROJECT OR GRANT NUMBER: If appropriate, enter the applicable research and development project or grant number under which the document was written.

8b. CONTRACT NUMBER: If appropriate, enter the applicable number under which the document was written.

9a. ORIGINATOR'S DOCUMENT NUMBER(S): Enter the official document number by which the document will be identified and controlled by the originating activity. This number must be unique to this document.

9b. OTHER DOCUMENT NUMBER(S): If the document has been assigned any other document numbers (either by the originator or by the sponsor), also enter this number(s).

10. DISTRIBUTION STATEMENT: Enter any limitations on further dissemination of the document, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this document from their defence documentation center."

(2) "Announcement and dissemination of this document is not authorized without prior approval from originating activity."

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document, even though it may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall end with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (TS), (S), (C), (R), or (U).

The length of the abstract should be limited to 20 single-spaced standard typewritten lines, 7½ inches long.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a document and could be helpful in cataloging the document. Key words should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context.